UNIVERSIDAD CARLOS III DE MADRID

Escuela Politécnica Superior

PROYECTO FIN DE CARRERA

MEMETRACKER – TDT :
Detección y seguimiento de temas en noticias y blogs

Presentado por:

Ignacio Nieto Vidaurrázaga

Dirigido por:

César de Pablo Sánchez

Agradecimientos

Resumen

El presente proyecto desarrolla un sistema de detección de tópicos y seguimiento de noticias llamado Tracker. Su funcionamiento permite asociar las noticias que recibe según su temática y contenido. Este sistema de seguimiento de noticias está adaptado para su funcionamiento sobre la base de datos Politiktracker, que contiene noticias de carácter político.

Dada la cantidad de información que se maneja hoy en día en los medios de comunicación y especialmente en la web, resulta imprescindible acudir a fuentes de información fiables, filtrar los datos que se obtienen (en forma de noticias, reflexiones u opiniones), para más tarde tratar toda esa información de la manera más adecuada y sacar el máximo provecho de su contenido. Realizar todo este proceso sin ayuda de procesos automáticos es *imposible* puesto que las noticias reflejan hechos de actualidad y por tanto el factor tiempo es vital si no se quiere restar actualidad a las noticias.

Dentro del ámbito de la información y más concretamente en la política está creciendo la importancia que se concede a las nuevas tecnologías y dentro de éstas juegan un papel decisivo los sistemas de detección y seguimiento TDT (Topic detection and tracking) en los que está comprendido el proyecto Tracker. En política interesa saber qué es nuevo, qué está sucediendo ahora mismo y qué temas tienen mayor calado entre los lectores. Todas esas cuestiones son abordadas mediante el desarrollo del sistema y resueltas tras su ejecución.

Abstract

This project deals with a system so-called Tracker, which enables gathering of topics and monitoring of news. For instance, Tracker links news received pursuant to the matter and content thereof. This system for the tracking of news is tailored for its functioning upon the MEMETRACKER database, made up of political content news.

Given the amount of information managed nowadays by media, specially through the Internet, it is crucial to turn to trustful sources of information as well as to filter data obtained therein (whether news, thoughts or opinions), so that said information can thereafter be accurately processed in views to make the most of it. Achievement of the aforementioned without help of automated means may be particularly onerous: indeed, relevant news reproduce current information and, thus, time is essential to avoid news being outdated.

Performance of new technologies is growing increasingly within the information scenario, namely as regards political news, especially those technologies allocated to topic detection and tracking systems (TDT), amongst which Tracker. "What is new?", "What is happening currently?" and "What topics are of the interest of readers?" are issues often raised in connection to politics. The latter concerns are tackled through the development of the system and cleared up after execution thereof.

Acrónimos

API	Application Programming Interface (Interfaz de Programación de Aplicaciones).
BD	Base de Datos
IA	Inteligencia Artificial
IDE	Entorno de Desarrollo Integrado
JDK	Java Development Kit
PLN	Procesamiento del Lenguaje Natural
RI	Recuperación de la Información (en inglés IR, Information Retrieval)
RSS	Really Simple Syndication
SE	Standard Edition
SO	Sistema Operativo
TDT	Topic Detection and Tracking (Detección y Seguimiento de Tópicos)
TREC	Text Retrieval Conferences (Conferencias sobre RI)
US	Umbral de Similitud
XML	Extensible Markup Language (lenguaje de marcas ampliable)

Índice general

Índice de figuras

Capítulo 1

Introducción

1.1. Motivación

La irrupción de la WWW (World Wide Web) como medio de comunicación, junto con el aumento de las capacidades de los ordenadores, ha dado lugar a que existan grandes repositorios de documentos que se modifican e incrementan a lo largo del tiempo. El acceso a este tipo de información generalmente se realiza mediante sistemas de Recuperación de Información (RI) o lo que más coloquialmente se denominan buscadores. Pero debido a la cantidad de información presente en los documentos es difícil localizar la información que se desea, sobretodo cuando queremos información sobre algún suceso particular.

En concreto, en el mundo de la política, y dado que la información es poder, es de extrema importancia tener acceso a la información deseada en el momento adecuado y poder filtrar la información deseada.

Y no sólo eso, también es vital tener mecanismos para evaluar las relaciones existentes entre los diversos temas a los que se tiene acceso para tener una noción de la estructura de los medios de comunicación digitales en cada momento. Ya no basta con almacenar la ingente cantidad de información que recibimos de los medios en cada momento, sino que es necesario manejar esa información para sacarle partido y no terminar siendo manejados por la información.

Para el proyecto general de la UC3M en el que se esta incluido el presente, se necesitaba implantar un sistema de detección y seguimiento de tópicos (TDT) que permitiera dar soporte a los usuarios interesados en conocer la actualidad política reflejada en los medios digitales de habla hispana. Ahí nace el proyecto *tracker* con la vocación de ser un sistema que proporcione un análisis de la información lo más profundo y útil posible.

1.2. Descripción general del proyecto “MEMETRACKER”.

El proyecto MEMETRACKER tiene como función constituirse en instrumento que proporcione una visión global del mundo de la política y el debate que se genera en torno a él.

Dada la ingente cantidad de información con la que bombardean los medios de comunicación, es necesario filtrar la información política utilizada para conseguir que los usuarios del sistema mejoren su comprensión de la misma. Y no sólo filtrarla son seleccionar única y exclusivamente aquella información que es acorde a sus intereses.

La aplicación que se pretende desarrollar tendrá en cuenta todos los aspectos relevantes del debate político haciendo especial hincapié en tres aspectos fundamentales de la información: los actores del debate (principales representantes políticos), las estructuras que conforman el entramado político (partidos políticos y agrupaciones), y por último los temas de especial candencia en las disputas políticas.

Para conseguir los resultados que se proponen, el sistema debe extraer la información de medios fiables, representativos y contrastados. Por ello, el sistema MEMETRACKER selecciona la información utilizando como fuentes los archivos de sindicación RSSy Atom provenientes tanto de los medios de información tradicionales (periódicos) como de los procedentes de la Web 2.0 (blogs, wikis, etc.) y sus comentarios asociados.

Como medio de presentación de la información política, el sistema proporcionará a los usuarios un interfaz gráfico mediante el cual se pueda acceder a la información en tiempo real y navegar por ella en profundidad.

1.3. Finalidad del proyecto “MEMETRACKER-TDT”.

Desde una perspectiva global este proyecto forma parte de otros tantos que se están desarrollando paralelamente y otro finalizado por parte de alumnos de Ingeniería Técnica en Informática de gestión en la Universidad Carlos III de Madrid. La cantidad de información disponible crece a un ritmo muy elevado y el valor de los datos como activo de las organizaciones está ampliamente reconocido. Para que los usuarios obtengan el máximo rendimiento de sus enormes y complejos conjuntos de datos son necesarias herramientas que simplifiquen las tareas de administrar los datos y de extraer información útil en el momento preciso. En caso contrario, los datos pueden convertirse en una carga cuyo coste de adquisición y gestión supere ampliamente el valor obtenido de ellos.

El proyecto MEMETRACKER-TDT tiene como fin el análisis de la noticias de la base datos del proyecto general MEMETRACKER para relacionarlas temáticamente. A partir de las relaciones establecidas por la aplicación, se tratará de determinar cuales son

los valores más adecuados para los parámetros de la aplicación y se estudiará también la variación de esos parámetros y su efecto sobre el funcionamiento de la misma.

Se trata de un analizador de texto que sirve para continuar con el tratamiento de la información política que comenzó el *Crawler* descargando a la base de datos las noticias políticas de blogs y páginas de Internet, y continuó el módulo de administración de la BD Politiktracker.

A continuación se muestra una imagen en la que se aprecia claramente como se integrará El ***tracker*** o **módulo de detección y seguimiento temático** dentro del proyecto MEMETRACKER con el resto de proyectos. La aplicación es una de las que producen resultados para el macroproyecto junto al módulo de popularidad y al módulo de procesamiento del lenguaje, ya que maneja los datos que le ofrecen otros módulos y extrae conclusiones que en este caso son las relaciones entre las diferentes noticias.

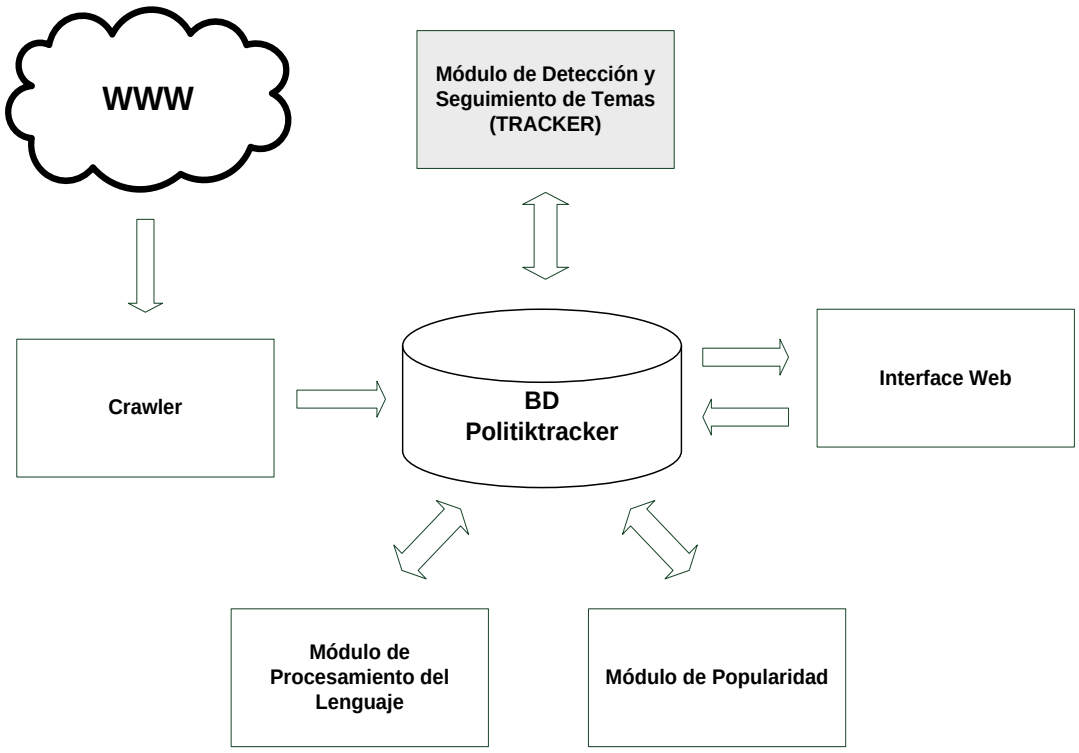


Figura 1.1: Estructura general del proyecto MEMETRACKER.

Se puede decir que el *Crawler* y el *administrador* de la *BD Politiktracker* se encargan de hacer disponible la información que se va a analizar (noticias políticas), el módulo *TRACKER* junto a otros dos hace posible la extracción de resultados en forma de conclusiones sobre la información, y por último el *Interface Web* muestra los resultados al usuario del sistema general.

La funcionalidad que ofrece dicho sistema a grandes rasgos es la siguiente: Un sistema *Crawler* ya desarrollado, conectado a Internet recopila información provenientes de

blogs y otros canales. Esta información se almacenará y relacionará entre si en la base de datos con el fin de llevar a cabo un seguimiento y una monitorización de la blogosfera hispana. En la citada base de datos se realizarán tareas de consulta y de adquisición de la información y otras relativas a la administración como pueden ser las copias de seguridad. Después de esto el *tracker* recoge la información de la base de datos, la trata relacionando noticias entre sí, e introduce en la BD la información de las relaciones producidas.

El *tracker* ofrece sus servicios a una estructura mayor como es el proyecto general pero la aplicación es de fácil adaptación a otros sistemas o entornos, sin embargo el sistema fue concebido como proyecto de investigación de ámbito universitario y de momento no tiene perspectivas de aplicarse a un entorno distinto.

Objetivos

El objetivo del proyecto es el desarrollo de un aplicación que permita la interacción con una base de datos para el tratamiento de la información que contiene. En concreto, busca relacionar noticias de la blogosfera hispana que están contenidas en la BD y luego dejar constancia de las relaciones temáticas en la BD. Al ser una tarea con cierto componente subjetivo será importante establecer mecanismos para valorar el funcionamiento de la aplicación. Para ello se analizará su funcionamiento en un experimento controlado a fin de valorar los resultados obtenidos y poder realizar mejoras en el futuro.

MEMETRACKER - TDT tiene como objetivo el desarrollo y administración de un sistema de detección y seguimiento de temas que trabajará sobre la base de noticias contenida en la BD Politiktracker. Este proyecto se encuadra en un objetivo más general como es la monitorización de la información contenida en blogosfera hispana. Además de la detección y seguimiento de temas, se plantea también como objetivo la evaluación de la manera en que se realiza el proceso para poder mejorarlo en un futuro.

En la siguiente lista se enumeran ls objetivos más concretamente:

- Realizar un estudio exhaustivo del dominio del problema a resolver que permita determinar las necesidades de gestión de la información.
- Establecer un diseño del sistema que combine eficacia en la consecución de los objetivos.
- Elegir las tecnologías necesarias para llevar a cabo el desarrollo del proyecto.
- Desarrollar la aplicación buscando una alta eficiencia que permita el mayor ahorro de recursos.

- Buscar la máxima modularidad posible para permitir cambios futuros en el sistema y un mejor entendimiento de cada parte del mismo.
- Realizar el desarrollo posibilitando una futura portabilidad de la aplicación a otros entornos diferentes del actual para el que está concebida.
- Conseguir el mayor rendimiento de los módulos externos a la aplicación, modificando los parámetros necesarios.
- Una vez terminado el desarrollo, realizar ejecuciones de la aplicación que sean representativas y que permitan un análisis de su funcionamiento.
- Evaluar el funcionamiento de la aplicación y extraer conclusiones.
- Proponer y establecer puntos de partida para posibles mejoras.
- Paralelamente con los objetivos anteriores también se busca una comunicación fluida con el resto de componentes del macroproyecto MEMETRACKER para mejorar el desarrollo del proyecto.

1.4. Estructura del documento

Esta memoria está dividida en 10 capítulos y un anexo y se estructura como sigue:

Capítulo 2: se hace una descripción de las tecnologías utilizadas en el desarrollo de la aplicación, y también se exponen las razones por las cuáles se utilizan estas tecnologías.

Capítulo 3: realizaremos una breve introducción al seguimiento de noticias tratando el elemento principal: la noticia.

Capítulo 4: se analiza la información relacionada con el tema del proyecto especialmente en el área de los Sistemas de Recuperación de Información (RI), de los Sistemas de Extracción de Información (EI) y de los sistemas de Detección y Seguimiento de Temas (TDT).

Capítulo 5: expone el diseño del algoritmo de tracking que se encargará de relacionar las noticias por temas, y muestra sus componentes y su funcionamiento.

Capítulo 6: mostrará el análisis y diseño de la aplicación, explicando cada componente de cada módulo de la aplicación.

Capítulo 7: se realiza la experimentación con el sistema para proceder a su evaluación. En primer lugar se establecen experimentos cualitativos para estudiar el proceso de relación de noticias, y después se pasa a los experimentos cuantitativos para determinar el resultado del proceso.

Capítulo 8: muestra las conclusiones a las que se ha llegado tanto en el proceso de desarrollo del sistema como tras la ejecución del mismo.

Capítulo 9: se delimitarán las líneas futuras con las que se pretende establecer las posibles pautas para mejorar las aplicación Y también se exponen las líneas de investigación que han quedado abiertas.

Capítulo 10: enumera los libros y recursos que han sido utilizados en alguna de las fases de desarrollo del proyecto.

Anexo A: contiene los corpus de noticias de los experimentos del capítulo 7. En concreto, este anexo dispone las tres colecciones de noticias necesarias para realizar los seis experimentos expuestos.

Capítulo 2

Estado del Arte

2.1. Introducción

A partir de los años 40, gracias a la evolución de los ordenadores, comenzó a almacenarse cada vez más información en formato digital, haciéndose cada vez más notable la necesidad de sistemas para mejorar el acceso a la información almacenada, con sistemas más rápidos y eficientes. Aunque los sistemas operativos poseen comandos para la búsqueda de ficheros de texto a partir de una consulta que posee una cadena de caracteres, (ej. 'grep de Unix'), estas aplicaciones son muy lentas a la hora de procesar grandes cantidades de texto de gran longitud, y además estos comandos son programas que simplemente buscan los ficheros que contienen la secuencia de caracteres de la consulta, no reconocen la información presente en los documentos.

Uno de los primeros sistemas de procesamiento de documentos fueron los sistemas de catálogos digitales para bibliotecas. En estos sistemas sólo se almacenaban los datos de las fichas existentes en las bibliotecas para facilitar la búsqueda de libros, como: el título, el autor, el año de publicación, un resumen, etc. Los sistemas de catalogo digital procesaban las fichas almacenadas en ficheros de texto permitiendo la búsqueda de libros en el catálogo especificando el texto que debe poseer cierto atributo del libro. La diferencia básica de los sistemas de catálogos y los sistemas de procesamiento de documentos actuales radica principalmente en la posibilidad actual de almacenar los documentos completos y no sólo ciertos atributos. Generalmente en los documentos de un sistema de procesamiento de documentos se añade una cabecera estructurada con datos acerca de la información que contiene el documento, a los cuales se denomina metadatos. Generalmente si los documentos son libros, los metadatos, suelen ser los atributos de las fichas en los catálogos. Los documentos generalmente poseen una estructura, que no siempre se halla explícita en el formato digital, pero que permite que los sistemas de procesamiento de los documentos puedan centrar sus consultas en partes del documento.

Mientras en las Bases de Datos Relacionales (BDR) se busca un campo que contenga exactamente las palabras de la consulta, en los sistemas de procesamiento de documentos se busca que las palabras de la consulta existan en cualquier parte del texto se realiza lo que llamaremos un emparejamiento entre cada una de las palabras de la consulta y del texto. Además, los campos de las BDR tienen un tipo de datos fijo, y en el caso de que sea texto, éste posee una longitud determinada, mientras que en los sistemas de procesamiento de documentos los campos de búsqueda (metadatos del documento o partes de un documento) son textos sin una longitud predeterminada.

Entre las técnicas de procesamiento de documentos cabe destacar las de recuperación, routing, filtrado, interpretación, clasificación, creación de resúmenes o de etiquetado automático de los documentos. Cada una de estas técnicas nos permite obtener un tipo distinto de información de la colección de documentos, aunque generalmente las aplicaciones están compuestas por varios de ellos. Estas aplicaciones se pueden clasificar básicamente en dos familias: los sistemas de Recuperación de Información (RI) y los sistemas de Extracción de Información (EI). La tarea principal de un RI consiste en buscar los documentos relevantes dentro de la colección que más se asemejen a la consulta del usuario y devolver estos al usuario. Si lo permite el sistema, los documentos recuperados se ordenarán según un valor de relevancia establecido por el sistema. Un sistema de EI procesa los documentos de una colección para extraer información estructurada específica. No intenta entender todo el documento, sino que analiza aquellas porciones de cada documento que contienen información relevante según unas pautas predefinidas. Los sistemas de procesamiento de documentos suelen aplicar técnicas tanto de un sistema de RI como de un sistema de EI. Por ello, en muchos casos es difícil clasificar una aplicación en uno de estos sistemas. Un método para su clasificación es el estudio de la funcionalidad del sistema de procesamiento de documentos: en un sistema de RI cada fichero o documento se ve como una secuencia de posibles palabras o términos significativos, y en un sistema de EI, cada fichero contiene frases o cláusulas significativas relevantes a un tema particular.

Al hallarse los documentos escritos en Lenguaje Natural se intentó inicialmente, aunque con poco éxito, aplicar en los sistemas de procesamiento de documentos las técnicas de Procesamiento de Lenguaje Natural (PLN). Actualmente estas técnicas son básicas en los sistemas de EI, y su uso en los sistemas de RI cada vez se está extendiendo más, ya que se pueden aplicar bien en el lenguaje de consulta del usuario, como en la representación y clasificación interna de los documentos.

En 1997 surgió una nueva iniciativa en el campo de los sistemas de RI, los sistemas de seguimiento y de detección de sucesos en noticias de actualidad. A esta iniciativa se le denominó Topic Detection Tracking (TDT) y está muy ligada al propósito del proyecto Tracker de obtener distintos temas (que agruparan un conjunto de noticias) sobre los sucesos que se relatan en los periódicos y otros medios de comunicación.

En el siguiente apartado se presentarán los sistemas de RI actuales para el procesamiento de grandes colecciones de documentos, las principales técnicas de búsqueda que se utilizan, y los sistemas de evaluación de estos sistemas (que serán desarrollados con más profundidad en el la sección 7 de esta memoria). En la sección 4.4 veremos los distintos algoritmos de agrupamiento de los documentos. En la sección 4.5 se estudiarán los sistemas que nos permiten obtener cierta información

predeterminada de los documentos mediante un procesamiento automático de éstos, o sea los sistemas de EI. Y en la sección 4.6 se profundizará en unos sistemas de RI centrados en la detección de sucesos, denominados TDT (parte central del proyecto).

2.2. Sistemas de Recuperación de la Información

Un sistema de Recuperación de Información (RI) es un sistema de procesamiento de documentos que trata de recuperar de una colección de documentos aquellos que se asemejan más a la consulta del usuario. Un sistema de RI se encarga tanto de la recuperación de documentos como de su almacenamiento y organización, al igual como ocurre en los Sistemas de Gestión de Bases de Datos. Pero hay que tener en cuenta que mientras en estos sistemas, los datos están totalmente estructurados, organizados expresamente para ser recuperados por el gestor de la base de datos, en los sistemas de RI los datos pueden ser de cualquier tipo y de cualquier longitud. En un SRI, un dato de consulta puede ser el documento completo, una parte del documento (puede ser una estructura del documento si este está estructurado o simplemente un párrafo del documento), o un metadato de la cabecera del documento, si este posee una cabecera estructurada. Si posee una estructura, se puede consultar en ciertos campos que representan las distintas secciones del documento. En cualquiera de estos campos se puede tener cualquier tipo de datos, no previamente especificados, y en el caso de datos de tipo texto pueden ser de cualquier longitud.

Un sistema de recuperación de datos simplemente comprueba si un dato existe en un fichero, es decir, busca documentos que casan con una palabra, mientras en los sistemas de RI se seleccionan aquellos documentos que coinciden parcialmente (búsqueda aproximada) o totalmente (búsqueda exacta) con los términos de la consulta del usuario, creándose una lista de documentos que en el caso de búsqueda aproximada se puede ordenar según un índice de relevancia entre la consulta y el documento.

El índice de relevancia de un documento con respecto a una consulta se calcula con una medida de semejanza que devuelve un valor en función de los términos de la consulta que se encuentran en el documento. Generalmente en los sistemas de búsqueda aproximada se fija un umbral de semejanza (en la aplicación TRACKER se denomina “Umbral de Similitud”) que depende de la colección. Cuando la relevancia de un documento respecto a una consulta está por debajo de este valor se supone que el documento no es relevante para el usuario por lo que no se incluye en la lista de documentos relevantes.

Las aplicaciones directas de estos sistemas son los servicios de información: catálogos, bibliotecas digitales, buscadores Web, enciclopedias, ofimática, documentación (patentes, leyes, bibliografía), sistemas multilingües o bien sistemas de integración y distribución de noticias, etc. Indirectamente estos sistemas ayudan a la construcción de léxicos (corpus, ontologías, bases de conocimiento, diccionarios, tesauros) y a la clasificación de documentos.

La mayoría de sistemas de RI permiten que las consultas se expresen como una lista de palabras que el usuario espera encontrar en el documento y que para él lo caracterizan. Algunos sistemas permiten la utilización de operadores (principalmente boléanos), palabras o cadenas con comodines. Sin embargo pocos sistemas incluyen la búsqueda

por estructura del documento puesto que esto requiere que el usuario conozca de antemano la estructura del documento, y además como cada colección de documentos posee su propia estructura, la tarea de reconocimiento de las distintas estructuras posibles añade más complejidad a los sistemas de indexación y recuperación.

Las conferencias TREC (Text Retrieval Conferences) están centradas en el desarrollo de los sistemas de RI. En ellas se ha comprobado que su efectividad está muy ligada a la consulta que hace el usuario y a la elección de los términos de indexación de los documentos. Por ello en las investigaciones actuales se está trabajando con el propósito de llegar a la utilización del Lenguaje Natural como lenguaje de consulta, de modo que se está investigando en los siguientes aspectos: la ayuda al usuario para la realización de las consultas por medio de tesauros, la expansión de consultas, y la realimentación automática de consultas. La gran cantidad de documentos almacenados actualmente en los ordenadores y la longitud de éstos son los dos grandes retos de los sistemas RI actuales, que dan lugar a que las aplicaciones desarrolladas teóricamente y probadas en pequeñas colecciones de documentos cortos, al aplicarse a los grandes repositorios de documentos completos existentes en la actualidad, no den los resultados esperados. Esta problemática se está estudiando en las últimas conferencias TREC.

2.2.1. Arquitectura de un sistema de RI

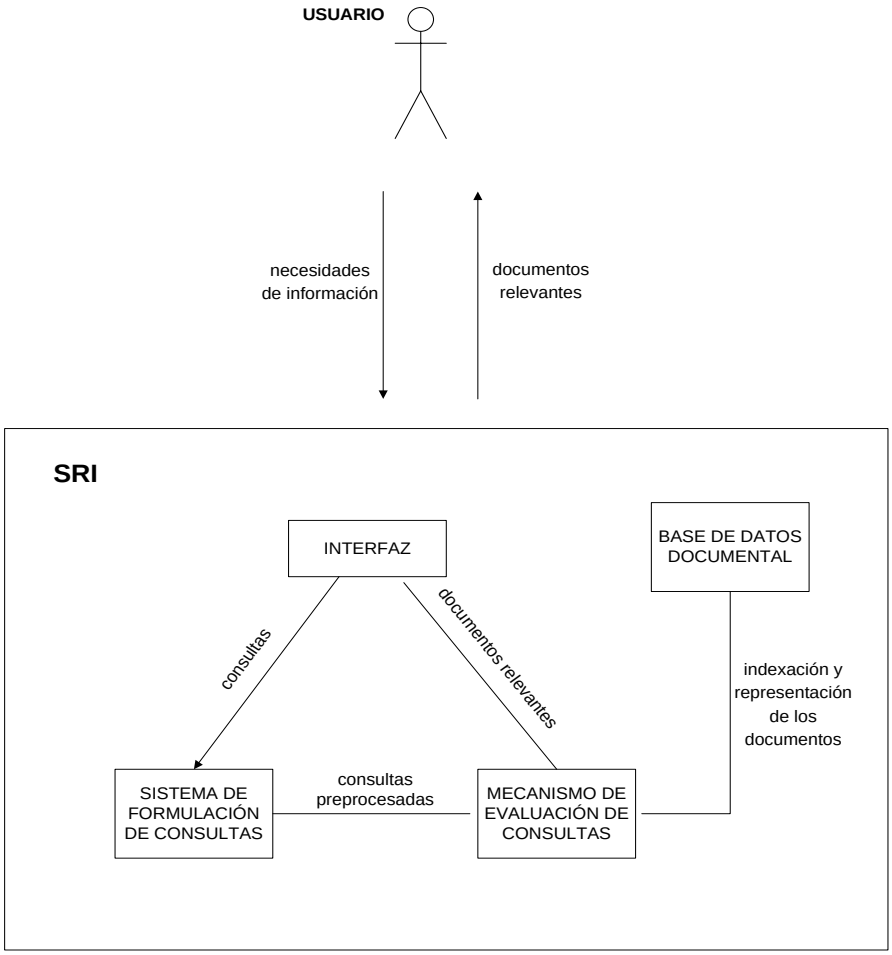


Figura 2.1: Componentes básicos de un sistema de recuperación de información.

Los sistemas de RI tradicionales utilizan programas que gestionan ficheros invertidos, ya que éstos permiten campos textuales de cualquier longitud, e incluso documentos multimedia. Cabe destacar que las bases de datos actuales se pueden utilizar para gestionar documentos. Un sistema RI generalmente se compone de:

- Una colección de documentos. Los documentos pueden ser estructurados y/o tener una cabecera estructurada con metadatos sobre el documento.

- Un lenguaje de consulta. Una consulta suele ser una lista de términos (palabras, combinación de palabras, raíces de palabras, cadenas con comodines) que pueden estar relacionadas con operadores ponderados según un nivel de relevancia.

- Una aplicación que se encargue de la organización, selección y presentación de documentos. Esta aplicación está a su vez compuesta de:

    - Un gestor del repositorio con un sistema de indexación de los documentos.

    - Un proceso de emparejamiento de la consulta con cada uno de los documentos, que devuelve la semejanza entre cada documento y la consulta.

    - Un proceso que organice los documentos relevantes, y los represente en una lista para que el usuario pueda extraer de esa lista los documentos que desea.

2.2.1.1. Sistema de indexación

Generalmente los sistemas de RI indexan los documentos para su posterior búsqueda. El método de indexación para textos completos más utilizado se basa en el uso de ficheros invertidos. Un fichero invertido contiene como entradas todos los términos significativos de la colección de documentos, y para cada término se indica en qué documentos aparece, pudiendo incluir ciertos atributos adicionales como la frecuencia de aparición o la posición de esa palabra en el documento. Un término puede bien ser una palabra, un lema o un sintagma nominal.

Los sistemas de búsqueda avanzada, permiten realizar consultas indicando la posición de los términos en las distintas partes del documento, o sea, campo, subcampos, párrafo o línea donde aparece la palabra en el documento. Todas estas propiedades se deben añadir al fichero invertido. Si el sistema permite además especificar la proximidad de los términos en el texto, se debe indicar en el fichero la posición relativa de cada palabra en el documento. Si el sistema permite realizar búsquedas por contenido, el índice no sólo tiene por cada entrada las apariciones de esa palabra sino también ofrece las apariciones de los términos similares. Para permitir la consulta de los términos que aparecen en ciertos metadatos, en las entradas de cada término se debe indicar las distintas apariciones del término en cada metadato.

Otro sistema de indexación muy utilizado son los ficheros de signaturas. Estos ficheros que contienen patrones de bits, denominados signaturas, que representan documentos. Un estudio comparativo de los dos métodos varias colecciones de documentos, ha demostrado que en realidad los ficheros de signaturas no mejoran los sistemas de RI, ya que los ficheros de signaturas producen más errores de emparejamiento de los esperados y el índice puede resultar mayor y más costoso de construir y actualizar.

2.2.1.2. La estructura de los sistemas de RI

En las colecciones de documentos completos, cuando se calcula el nivel de semejanza, éste a veces no refleja lo que busca el usuario. Esto es debido a que cuando un documento es muy largo, generalmente se habla de más de un suceso o temática. Generalmente hay un tema o suceso principal e información relacionada. Por ello se está estudiando técnicas que permitan centrar la búsqueda en pasajes, utilizando la estructura del texto. De este modo se calcula la semejanza para cada pasaje y se combinan para calcular la relevancia global.

La obtención de los pasajes será sencilla si la estructura del documento es explícita (si el texto posee etiquetas que limiten secciones del texto) como es nuestro caso con las etiquetas que contienen los archivos RSS.

2.2.2. Modelos de Recuperación de Información

Uno de los problemas más importantes en los sistemas de RI es cómo discernir entre los documentos relevantes o no. Los sistemas de RI clásicos utilizan técnicas de búsqueda booleanas y de reconocimiento de patrones. Estas técnicas llamadas de búsqueda exacta son técnicas muy restrictivas, ya que no tienen en cuenta las ambigüedades que aparecen en el Lenguaje Natural y generalmente los usuarios tienen problemas en la construcción de las consultas. Esto provoca que los usuarios no obtengan los resultados deseados, o bien no recuperan suficientes documentos, o recuperan tantos que es imposible comprobar cuáles de ellos son relevantes.

Debido a los problemas que plantean los sistemas de búsqueda exacta se han desarrollado técnica de búsqueda basadas en la información estadística, las denominadas técnicas de búsqueda aproximada. El sistema de búsqueda aproximada no hace un emparejamiento exacto y permite que el usuario especifique la importancia de cada uno de los términos. El sistema calcula la relevancia en función de los términos de la consulta que aparecen en el documento. Si la relevancia supera un cierto umbral, el documento se recupera, aunque alguno de los términos de la consulta no exista en el documento, La realización de las consultas resulta más compleja en estos sistemas.

2.2.2.1. Sistemas de RI de búsqueda exacta

En los sistemas de RI de búsqueda exacta se utiliza una correspondencia exacta entre los términos de las consultas y de los documentos. Cuando se realiza correspondencia entre una consulta y un documento, el sistema le asigna un valor de relevancia de ‘1’ al

documento, mientras que si no se realiza el emparejamiento tiene relevancia ‘0’. Así pues el sistema devuelve al usuario una lista con todos los documentos con relevancia ‘1’. En este sistema el primer documento de la lista no tiene por que ser más interesante para el usuario, simplemente es el primero que ha encontrado relacionado con la consulta. Es decir, todos los documentos de la lista tienen la misma relevancia, no se tiene en cuanta la frecuencia de aparición, ni el orden o importancia de los términos de la consulta.

Destacan dos modelos de sistemas de RI por búsqueda exacta:

Por Búsqueda de Patrones. Los sistemas de RI por búsqueda de patrones utilizan técnicas de reconocimiento de patrones. En estos sistemas la consulta puede ser una colección de palabras, cadenas con comodines o expresiones regulares. El sistema intenta buscar los documentos que contengan el patrón de la consulta. Este sistema de búsqueda no requiere índices, puede utilizar los documentos de la colección directamente. No suele ser muy útil en colecciones grandes por ser muy lento, pero es muy útil en colecciones de documentos que se modifican frecuentemente.

Por Indexación Boolena.

Se trata de uno de los modelos de recuperación de información más simples que se conocen. Se fundamenta en el álgebra de Boole y en la teoría de conjuntos. Este modelo crea una expresión booleana para formalizar la consulta y utiliza los operadores booleanos AND, OR y NOT.

Dependiendo de los operadores booleanos que unan las palabras a buscar, se recuperarán unos documentos u otros, puesto que no es lo mismo buscar palabra1 AND palabra2 (tiene que aparecer ambas) que buscar palabra1 OR palabra2 (aparece o una o la otra).

El problema de este modelo es que si encuentra una serie de documentos, no sabe ordenarlos según la relevancia que tenga cada uno. Para solucionarlo se puede utilizar el modelo booleano extendido que añade pesos a las palabras buscadas lo que le lleva a aproximarse a un modelo vectorial.

	t_1	t_2	t_3	...	t_j	...	t_m
d_1	0	0	1	...	1	...	1
d	0	0	1	...	0	...	1
d_i	0	1	1	...	1	...	0
d	0	0	1	...	0	...	1
d_n	0	1	1	...	0	...	1

Figura 2.2: Matriz de términos – documentos del modelo booleano.

2.2.2.2. Búsquedas aproximadas

Los estudios sobre el modelo booleano dieron lugar a que estos modelos se ampliaran para no desestimar un documento porque en él no aparecieran todos los términos de la consulta. De esta manera nacieron los modelos de búsqueda aproximada que se detallan a continuación y entre los cuales hay que destacar el modelo vectorial por ser este modelo es el más conocido y uno de los más utilizados en sistemas de RI, además de ser el que utilizada en nuestra aplicación mediante la librería Lucene,

Modelo vectorial

También conocido como modelo de espacio vectorial, está basado en el modelo booleano, pero mejorado, de manera que se asigna a cada término de la consulta un peso que puede ser cualquier valor positivo (binario, entero o real). Dentro de este modelo los documentos son representados utilizando un vector en el que se recogen las relaciones existentes entre el documento y sus características. La consulta también se representa como un vector por lo que este modelo resulta perfecto para realizar la comparación entre documentos y consultas.

Para obtener las características que ayudan a la formación del vector, se utilizan las ocurrencias encontradas de algunas palabras significativas dentro del texto.

Con estos datos se realiza la representación vectorial que será usada en las consultas para recuperar la información. La forma de recuperar la información es comparando este vector con los vectores de los documentos. Se usa una función de similitud. El grado de similitud varía según la consulta que se realice. Cuanto mayor es el grado se considera que más se ajusta a la petición.

Los documentos se representarán mediante una matriz de frecuencia de términos, y una consulta se representará de la misma forma $d_k = (w_{k,1},...,w_{k,j},w_{k,m})$

	t_1	t_2	t_3	...	t_j	...	t_m
d_1	w_{11}	w_{12}	w_{13}	...	w_{1j}	...	w_{1m}
d_2	w_{21}	w_{22}	w_{23}	...	w_{2i}	...	w_{2m}
..	..	..	..	..	..	..	..
d_i	w_{i1}	w_{i2}	$w_{i3}1$	...	w_{ij}	...	w_{im}
..	..	..	..	..	..	..	..
d_n	w_{n1}	w_{n2}	w_{n3}	...	w_{nj}	...	w_{nm}

Figura 2.3: Matriz de frecuencias de términos del modelo vectorial.

En este modelo se proponen las siguientes propiedades para los términos:

- tf_{ij}: es la frecuencia de aparición del término t_j en el documento d_i.
- df_j: indica el número de documentos en los que aparece el término t_j.

A partir de éstas, el peso de cada término en el documento $w_{i,j}$, se calcula generalmente según la siguiente función:

- $w_{i,j} = tf_{i,j} \cdot idf_j$, donde idf es la función inversa de df.

Así pues, si se utiliza la medida del coseno, la semejanza entre un documento d_j y la consulta d_k siendo m el número de términos, viene dada por:

$$sim(d_j, d_k) = \frac{\sum_{i=1}^{m} w_{j,i} \cdot w_{k,i}}{\sqrt{\sum_{i=1}^{m} w_{j,i}^2 \cdot \sum_{i=1}^{m} w_{k,i}^2}}$$

Existen diferentes funciones para medir la similitud entre documentos y consultas. Todas ellas están basadas en considerar ambos como puntos en un espacio *n*-dimensional. Como ejemplo, se muestran algunas en la siguiente tabla:

Medida de Similitud	*Modelo Booleano*	*Modelo Vectorial*
Producto escalar	$\|X \cap Y\|$	$\sum_{j=1}^{m} X_j \cdot Y_j$
Coeficiente de Dice	$\frac{2 \cdot \|X \cap Y\|}{\|X\| + \|Y\|}$	$\frac{2 \cdot \sum_{j=1}^{m} X_j \cdot Y_j}{\sum_{j=1}^{m} X_j^2 + \sum_{j=1}^{m} Y_j^2}$
Coseno	$\frac{\|X \cap Y\|}{\sqrt{\|X\|} \cdot \sqrt{\|Y\|}}$	$\frac{\sum_{j=1}^{m} X_j \cdot Y_j}{\sqrt{\sum_{j=1}^{m} X_j^2 \cdot \sum_{j=1}^{m} Y_j^2}}$
Coeficiente de Jaccard	$\frac{\|X \cap Y\|}{\|X\| + \|Y\| - \|X \cap Y\|}$	$\frac{\sum_{j=1}^{m} X_j \cdot Y_j}{\sum_{j=1}^{m} X_j^2 + \sum_{j=1}^{m} Y_j^2 - \sum_{j=1}^{m} X_j \cdot Y_j}$

Figura 2.4: Distancias entre dos vectores de términos.

Con este modelo además se pueden obtener los documentos de forma ordenada y se puede limitar el número de resultados si se considera un grado de similitud mínimo.

Modelo Probabilístico

Como su mismo nombre indica, este modelo se fundamenta en el cálculo de la probabilidad de que el documento sea relevante para la consulta realizada. Por tanto si cogemos un documento cualquiera entre un conjunto de m documentos, existe una cierta probabilidad de que dicho documento sea relevante para la pregunta realizada. Se tienen que analizar las características que hacen a un documento ser relevante.

La fórmula para obtener la probabilidad de ser relevante, es decir la **relevancia** del documento es:

$$P(relevancia) = \frac{m}{N}$$

donde *m* es el conjunto de documentos relevantes y *N* es el conjunto de todos los documentos.

Para calcular la relevancia se utilizan una serie de pesos dados a las características del documento. Para saber la relevancia se usan índices de los términos que se conocen como descriptores con los pesos que se han establecido. Con esto se pretende recuperar los documentos en los que existen los mejores descriptores de los que el usa en la consulta.

Puesto que usa pesos, se puede calcular un determinado grado de relevancia y con el cual los resultados obtenidos pueden ser ordenados como sucedía en el modelo vectorial o en el booleano extendido.

El principal problema de este modelo es el hecho de que necesite una hipótesis para comenzar su ejecución y mediante la que se inicialicen los documentos relevantes así como los pesos. Además de esto como contabiliza el número de términos que aparecen y los supone independientes hace que todo el cálculo de estimación de probabilidades iniciales sea complejo.

Hay que reseñar que existen otros modelos importantes en lo que a Recuperación de la Información se refiere. No merece la pena profundizar mucho en ellos dada su complejidad y poca relación con el presente trabajo, pero son modelos muy potentes para todo sistema de RI. Algunos de estos modelos son:

- ⇒ **Basados en Modelos de Lenguaje**
- ⇒ **Basados en Redes de Inferencia**
- ⇒ **Basados en Lógica Difusa**
- ⇒ **Basados en Semántica Latente**

2.2.3. Evaluación de los sistemas de RI

Principales medidas de evaluación en RI.

Una vez definido el concepto de relevancia (en el modelo probabilístico) y relacionando éste con si un documento es recuperado o no, podemos establecer una serie de medidas que nos servirán para evaluar los sistemas de recuperación. A continuación expondremos las principales medidas comunes a todos lo modelos de recuperación.

Los documentos pueden ser recuperados o rechazados al establecer la comparación entre la pregunta y la base de datos. El conjunto de documentos recuperados se divide, salvo en los sistemas perfectos, en dos grupos: documentos relevantes recuperados, es decir aquellos que se han recuperados correctamente y los no relevantes, recuperados erróneamente que provocan ruido en la salida. Los documentos no recuperados, que a su vez se dividen en los relevantes, rechazados por el sistema de manera errónea y los no relevantes, rechazados de manera correcta por el sistema. Esto mismo lo podemos ver en el siguiente dibujo.

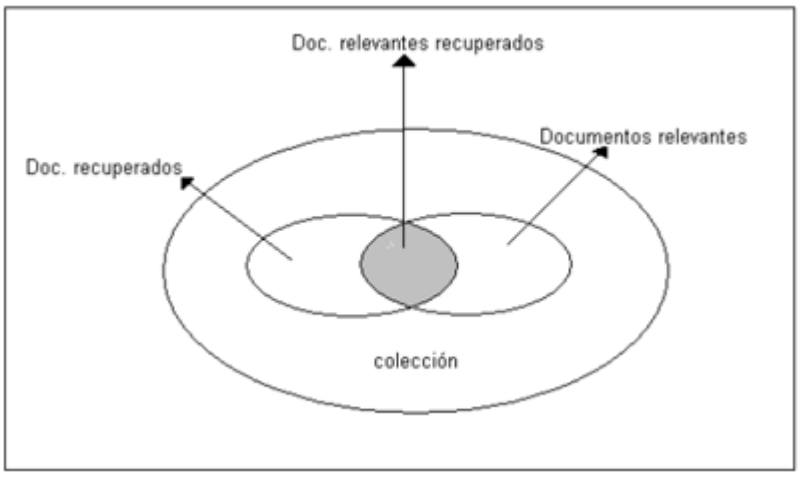


Figura 2.5: Esquema de recuperación de documentos.

Para la representación matemática de las medidas distinguiremos entre documentos recuperados (B y C) y no recuperados (A y D); y también, entre documentos relevantes (C y D) y no relevantes (A y B):

Figura 2.6: Esquema de división de documentos.

Precisión

La precisión es la proporción de documentos relevantes recuperados sobre el número total de documentos recuperados, siendo su fórmula:

$$\text{Pr}\,ecisión = \frac{C}{B+C}$$

Esta medida está relacionada con dos conceptos, el de ruido y el de silencio informativo. De este modo, cuanto más se acerque el valor de la precisión a 0, mayor será el número de documentos recuperados que no le sirvan al usuario y por lo tanto el ruido que encontrará será mayor.

La salida obtenida en la recuperación es ordenada en función de la relevancia, por lo que los documentos más relevantes están al comienzo de la salida, de esta manera a medida que avanzamos en el número de documentos recuperados, la precisión decae.

Exhaustividad o cobertura

La exhaustividad es la otra medida principal utilizada en la evaluación de los sistemas de recuperación. Es la proporción de documentos relevantes recuperados en una búsqueda determinada sobre el número de documentos relevantes para esa búsqueda en la base de datos, y su fórmula es:

$$Exhaustividad = \frac{C}{C + D}$$

Muchos autores, por influencia del término inglés la denominan "*recall"* o *"rellamada"*. Es la proporción de material relevante recuperado, del total de los documentos que son relevantes en la base de datos, independientemente de que éstos, se recuperen o no. Esta medida es inversamente proporcional a la precisión

Si el resultado de este cálculo tiene como valor 1, tendremos la exhaustividad máxima, ya que hemos encontrado todo lo relevante que había en la base de datos, por lo tanto no tendremos silencio informativo: la recuperación será perfecta en lo que se refiere a cobertura de las noticias (no en cuanto a la precisión).

Una de las características de la cobertura o exhaustividad es que mientras que la precisión se puede determinar, la exhaustividad no, ya que para calcularla necesitamos previamente el número de documentos relevantes (es el trabajo previo a la ejecución del TRACKER, realizando los grupos de noticias o *clusters* manualmente).

Relación entre la precisión y exhaustividad

Necesitamos comprobar que la precisión y la exhaustividad están compensadas, ya que un sistema con una exhaustividad muy alta pero con baja precisión y viceversa no será adecuado. Para comprobar como se relacionan la precisión y la exhaustividad en una

sola gráfica, podemos hacerlo de varias maneras: calculando la *precisión exhaustividad interpolada*: es decir tomamos un conjunto de documentos y calculamos para cada valor de precisión su exhaustividad. Por ejemplo tomamos los veinte primeros documentos recuperados, donde hay quince documentos relevantes (traducido a la aplicación TRACKER se trata de 15 documentos sobre un mismo tema) y calculamos la precisión y la exhaustividad para cada documento recuperado (si el primer documento recuperado es relevante tendremos una precisión de 1/1 y una exhaustividad de 1/15).

Una vez que tenemos estos valores, en ambos casos marcamos los puntos, en el eje de las *x* los valores correspondientes a la exhaustividad y para cada valor de ésta marcamos en el de las *y* el valor de la precisión que le corresponde. Uniendo los puntos obtenemos la curva que nos dice cómo se relacionan estas dos medidas en cada sistema y comparándolas ver qué sistema es el más efectivo.

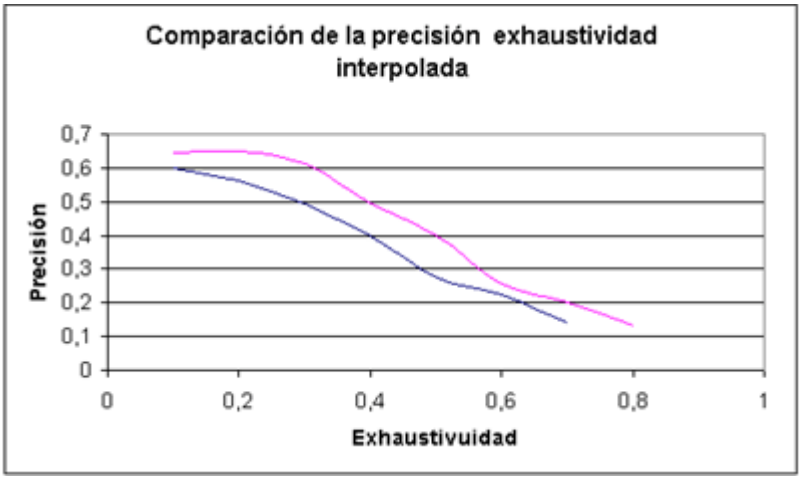


Figura 2.7: Comparación de la precisión y la exhaustividad.

Como se puede apreciar, la precisión disminuye cuando la cobertura aumenta. La relación entre ambas variables es inversamente proporcional pero no tiene por qué ser lineal sino que la precisión puede aumentar exponencialmente con el descenso de la cobertura y viceversa.

Según las necesidades del usuario se buscará una mayor precisión o cobertura. En concreto para el proyecto MEMETRACKER será conveniente que se consiga la mayor cobertura sobre los temas de las noticias.

La medida de f (F-Measure)

Una medida unificada de ambas, F, es calcular su media armónica para un número concreto *x* de documentos recuperados.

La fórmula viene dada por:

$$F - Measure = \frac{2}{\frac{1}{E(x)} + \frac{1}{P(x)}}$$

donde *E(x)* es la exhaustividad o cobertura para el conjunto de documentos recuperados *x*, y *P(x)* la precisión para el mismo conjunto de documentos.

Para darle más peso a alguna de las dos medidas, precisión o cobertura, se puede utilizar este otro valor de F-Measure:

$$F - Measure = \frac{1 + b^2}{\frac{b^2}{E(x)} + \frac{1}{P(x)}}$$

La diferencia con la anterior fórmula es "b", cuyo valor permite dar un mayor peso a la precisión, si es mayor que 1, o a la exhaustividad si es menor.

2.2.4. El Procesamiento del Lenguaje Natural en RI

En 1999 comenzó una discusión sobre las mejoras de la utilización de las técnicas de PLN en los sistemas de RI. En la actualidad la mayoría de investigadores están a favor de que la contribución de las técnicas avanzadas del PLN es en realidad pequeña y además la efectividad de los sistemas de RI está estrechamente relacionada con la formulación de las consultas. Si una consulta no está muy bien formulada, los errores que producen los sistemas de PLN a menudo pueden empeorar la eficacia de los sistemas de RI. Además, las consultas son difíciles de procesar si se tienen en cuenta todos los sinónimos, y los emparejamientos de palabras, lo que produce una explosión lingüística.

Pero ello no significa que los sistemas de PLN no sean útiles en RI, muy al contrario, se ha probado que las técnicas básicas de PLN, como la extracción de raíces, o técnicas más avanzadas como detección de multi-términos y nombres propios, detección de relaciones de sinonimia y expansión de consultas juegan un papel muy importante en los sistemas de RI.

Las líneas de investigación del PLN aplicado a los sistemas de RI son básicamente los siguientes:

- la interacción basada en el significado (búsqueda conceptual).
- respuesta a preguntas concretas (no búsqueda de documentos).
- creación automática de resúmenes como respuesta a las consultas.
- integración de información.

- creación de consultas altamente descriptivas, precisas y elaboradas.
- multilingüismo.

Las implementaciones lingüísticas básicas provenientes de los sistemas de PLN para mejorar los sistemas de RI son:

- Segmentación del texto en vocablos (Tokenizing). En algunos idiomas esta tarea es muy sencilla ya que existen separadores entre las palabras. Pero en idiomas como el japonés o el chino, para extraer las palabras del texto se requiere la ayuda de un diccionario y la utilización de patrones.

- Extracción de raíces. Se suele utilizar para unificar los términos con variantes morfológicas. La falta de un dominio específico o del conocimiento del contexto da lugar a que estos sistemas provoquen fallos en la recuperación (varias palabras con significados distintos a veces tienen la misma raíz).

- Utilización de listas de palabras de parada. El estudio de los sistemas de RI ha demostrado que la presencia de las palabras con poco valor semántico o demasiado frecuentes influyen poco en su efectividad. Por ello los indexadores de sistemas de RI, poseen colecciones de palabras de este tipo, de modo que cuando una palabra pertenece a esta lista no se indexa. Con ello se reduce el espacio de búsqueda y el tamaño del índice.

Algunos sistemas utilizan algunas implementaciones de técnicas de PLN un poco más complejos como son:

- Identificación de frases. El objetivo es utilizar frases como unidades de indexación. La creación de frases como unidades, se basa en presuponer que algo ha pasado con anterioridad en las frases procedentes, de modo que cuando hay elementos cohesivos entre varias sentencias se forma una unidad.

- Identificación de nombres de entidades. Estos pueden ayudar a identificar nombres propios, nombres de lugar y organizaciones. Para ello se aplican técnicas de análisis de patrones, a partir de la aplicación de reglas (manuales o creadas mediante un sistema de aprendizaje) o bien mediante modelos de Harkov (que requieren un conjunto de entrenamiento de documentos etiquetados).

- Extracción de conceptos. Es una versión más general de la extracción de nombres de entidades. Se intentan identificar nombres de ciudades, países, provincias, títulos, fechas, monedas, porcentajes, nombres químicos, etc. La obtención de esta información es similar a la de nombres de entidades, el problema a resolver es determinar cuándo se deben utilizar los conceptos y cuándo un sistema de RI los requiere. Otra cuestión por resolver es la normalización de conceptos y los problemas en su detección. Por ejemplo, si se desean extraer porcentajes, hay que tener en cuenta sus posibles representaciones ‘95%’, ‘0.95’, ’95 por ciento’. Esta tarea es básica en los sistemas de EI, y se intenta aplicar a los sistemas de RI para mejorar su eficacia. La extracción de conceptos requiere técnicas de:

- Desambiguación del significado de las palabras.
- Adquisiciones léxicas.
- Análisis de sentencias.
- Expansión de sinónimos.
- Resolución de anáforas.

De los experimentos realizados se concluye que la aplicación de técnicas de PLN básicas, así como las de identificación de frases y entidades, mejoran la efectividad de los sistemas de RI. Estas técnicas permiten que tanto en la indexación como en las consultas, se tengan en cuenta las variaciones morfológicas, la polisemia, las relaciones semánticas de sinonimia e híper/hiponimia, además de reconocer términos multi-palabras, dependencias terminológicas, colocaciones, agrupamientos, etc.

2.2.5. Sistemas de agrupamiento (Clustering)

Los sistemas de agrupamiento de documentos tienen la tarea de clasificar los documentos según sus propiedades intrínsecas en varios grupos (clusters). Mientras que en los sistemas de clasificación los documentos se clasifican según semejanza o relevancia con ciertas clases previamente especificadas, en los sistemas de agrupamiento se trata de buscar característica que permitan separar los documentos en grupos basándose en las propiedades internas de la colección. Idealmente los grupos deben estar completamente separados, pero algunas veces el solapamiento entre grupos es inevitable. El correcto funcionamiento de estos sistemas depende de las propiedades estadísticas de la colección. Generalmente estos sistemas se aplican en colecciones estáticas, aunque también se puede aplicar a colecciones que se incrementan en el tiempo.

Los algoritmos de agrupamiento se clasifican en dos grandes subgrupos, los algoritmos **jerárquicos** y los **no jerárquicos**.

- Los algoritmos no jerárquicos generan una partición del conjunto de documentos en un conjunto de grupos sin relaciones jerárquicas entre los grupos, utilizando distintas heurísticas. Entre estos algoritmos hay que destacar el utilizado para nuestra aplicación que es el siguiente:

    - **Single-pass.** Cada vez que llega un documento se compara con todos los grupos generados hasta el momento. Si ningún grupo se asemeja al documento, entonces el documento forma un nuevo grupo. Produce grupos dependientes del orden de procesado.

- Los algoritmos jerárquicos calculan la semejanza entre todos los pares de grupos y producen una secuencia anidada de particiones. La ventaja de estos algoritmos viene dada por la posibilidad de utilizar las técnicas de búsqueda en árboles para la resolución de consultas. Uno de los problemas de los algoritmos jerárquicos es la elección del número de grupos a obtener, ya que cuando la colección es grande (100 objetos), la representación utilizando jerarquías no suele ser muy apropiada. Dentro de estos algoritmos existen dos enfoques: el enfoque abajo-

arriba que comienza suponiendo que cada documento es un grupo individual y se unen iterativamente de dos en dos los grupos cuya función de semejanza supere un valor; el enfoque arriba-abajo, en el que se genera al principio un grupo formado por todos los documentos y progresivamente se van subdividiendo hasta conseguir tantos grupos como documentos.

2.2.6. Seguimiento, detección y clasificación de sucesos (TDT)

Los sistemas para el seguimiento y la detección de sucesos en las noticias (TDT – Topic Detection and Tracking) comenzaron en 1997 soportada por el DARPA dentro del programa TIDES (Translingual Information Detection, Extraction and Summarization).

Es dentro de estos sistemas dónde se encuadra la aplicación TRACKER ya que trata de detectar sucesos en una base de noticias (agrupando las noticias según los temas a os que se refieren). Su objetivo es la detección de temas y agrupación y seguimiento de las noticias.

En TDT se han definido cinco tareas de investigación:

1. **Segmentación de noticias** (Store Segmentation). Consiste en extraer cada noticia de la colección. Esta etapa es trivial en el caso de noticias escritas como los periódicos ya que el formato del texto permite detectar cuando empieza y termina cada noticia. Pero en las noticias habladas, como las radio-noticias y los telediarios, esta tarea es más compleja, siendo todavía una línea de investigación abierta.

2. **Seguimiento de sucesos** (Topic Tracking). Consiste en clasificar las noticias en un conjunto de sucesos predeterminados. El problema se resuelve con técnicas te clasificación supervisada donde el sistema conoce a priori los sucesos de interés. Para ello se posee una colección de noticias ya clasificadas en cada uno de los sucesos de interés para el entrenamiento del sistema.

3. **Detección de temas** (Topic Detection). Consiste en buscar los distintos sucesos que aparecen en las noticias y agrupar las noticias que hablan sobre el mismo tema. Este es un problema de clasificación no supervisada, donde se trata de organizar o agrupar automáticamente las noticias sobre el mismo suceso. En este caso, a diferencia del anterior, no hay documentos para entrenar el sistema, y además no se conocen a priori los sucesos de interés.

4. **Creación de temas** (First Story Detection). El sistema debe decidir si un documento representa un nuevo tema considerando todos los relatos del corpus. Por tanto, en este caso la tarea consiste en marcar cada nueva noticia si se trata de la primera noticia sobre un suceso o si por el contrario no es la primera noticia hablando del mismo.

5. **Enlazar noticias** (Store Link Detection). Trata de detector cuándo dos noticias hablan sobre el mismo suceso. El sistema debe comprender qué es un tema, independientemente de los temas específicos, y calcular la semejanza entre pares de documentos. Esta tarea no trata de dividir los documentos en conjuntos ortogonales, se permite que un documento hable de distintos temas, por lo que un documento puede pertenecer a varios grupos.

En los trabajos de TDT se ha comprobado que la utilización de técnicas que detecten en el relato las expresiones que hablan acerca de quién, qué, cuándo y dónde ocurre el suceso, permiten aumentar la efectividad de estos sistemas, ya que estas expresiones son básicas en la definición de un suceso. Sin embargo, hay que tener en cuanta que estas palabras pueden diferir a lo largo del tiempo en los relatos que tratan la misma historia, debido principalmente a la evolución del suceso, lo cual produce la inclusión en las noticias de nuevos sucesos muy relacionados, con más información y datos sobre el suceso.

Evaluación de los sistemas TDT

Los sistemas TDT son sistemas de recuperación de información utilizados para la detección de sucesos. Por ello y teniendo en cuenta que los sistemas TDT son una tipo de sistemas de RI para su evaluación se utilizan principalmente las mismas medidas, es decir, la Precisión, la Cobertura y la F-Measure que fueron definidas en el apartado 4.2.3.

Además en ocasiones se utilizan otras dos medidas para completar la evaluación que son la Tasa de Fallo, y la Tasa de Falsa Alarma

Si tenemos en cuenta la siguiente tabla de contingencia para evaluar los sistemas TDT:

	Relevante	No Relevante
Recuperado	a	b
No recuperado	c	d

Las nuevas medidas se definen como:

$$Tasa\_de\_Fallo = \frac{c}{a+c}$$

$$Tasa\_de\_Falsa\_Alarma = \frac{b}{b+d}$$

Capítulo 3

Herramientas utilizadas

3.1. Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora

software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no es software libre).

3.1.1. JDBC

Java Database Connectivity, más conocida por sus siglas **JDBC**, es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java, independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede, utilizando el dialecto SQL del modelo de base de datos que se utilice.

El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos (URL) que pueden manejar. Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la biblioteca de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión, para ello provee el localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar con cualquier tipo de tareas con la base de datos a las que tenga permiso: consulta, actualización, creación, modificación y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.

En el caso del Tracker, es necesario utilizar JDBC para realizar las operaciones con la base de datos Politiktracker ya que las noticias a analizar se encuentran almacenadas ahí. Con JDBC se realizará la conexión con la BD, se extraerán las noticias nuevas que le hayan llegado mediante consultas y serán analizadas por el Tracker. Posteriormente y cuando el proceso de relación de las noticias haya finalizado, vuelve a entrar en juego JDBC para permitir realizar inserciones sobre la tabla de la BD. Para terminar y siempre que el sistema haya terminado su cometido se debe cerrar la conexión.

3.2. MySQL

MySQL es el sistema gestor de base de datos de software más popular del mundo. Es un sistema de gestión de base de datos relacionales y de código abierto. Como sistema de gestión de base de datos está considerado entre los de mejor rendimiento (mayor velocidad), es multiproceso (funciona en subprocesos independientes), es multiusuario y de excelente fiabilidad. Al considerar a MySQL como sistema de gestión de base de datos damos a entender que cumple con la doble función de servidor y gestor de los datos.
Un servidor de base de datos relacionales mantiene la información en tablas independientes siguiendo un modelo de almacenamiento que permite un rápido acceso a los datos mediante el uso de un lenguaje de consulta denominado SQL (Structure Query Language).
MySQL AB fue la creadora de MySQL. Fue fundada en 1995 y es una de las grandes empresas de software libre del mundo. Recientemente, Sun Microsystems ha anunciado un acuerdo para adquirir MySQL AB. Entre los usuarios de MySQL AB se encuentran

Alcatel-Lucent, Amazon.com, Google, Digg, Facebook, Nokia, Yahoo y YouTube, entre otro.
A continuación se detallan una serie de características propias de MySQL. En este caso validas para la versión MySQL 5.0.51 que ha sido utilizada para la realización del proyecto.

LICENCIAS DE USO

En sus inicios MySQL fue software totalmente libre, posteriormente se implanto el criterio de la doble licencia, que es el criterio vigente.
MySQL sigue gozando del estatus de licencia pública general (GPL, General Public License) o GNU que autoriza su uso limitado a cualquier persona con la condición de que esta no lo redistribuya y si lo hiciese debe hacerlo con el mismo tipo de licencia. La licencia GPL GNU es sin cargo.
La licencia GPL GNU tiene precisas reglas de uso que se pueden consultar en: www.gnu.org/licenses, pero podemos resumir sus principios básicos en los siguientes:

- Libertad para ejecutar el programa de software libre.
- Libertad para acceder al funcionamiento de código interno.
- Libertad para modificar el código para adaptarlo a nuestros fines.
- Libertad para distribución de copias.

Muchos desarrolladores de software se benefician de que MySQL pueda distribuirse como GPL GNU pero no desean esto para su software. Para ellos se creó la licencia comercial.
Cuando un desarrollador de software no desea que su software sea GPL GNU pero quiere aprovechar la funcionalidad MySQL en sus aplicaciones tiene la opción de adquirir una licencia comercial.
Se debe tener en cuenta que MySQL en su versión GPL no tiene ninguna garantía, pero SUN se responsabiliza por el producto contratado con la licencia comercial.

Por todas estas ventajas se ha elegido MySQL como sistema gestor de MEMETRACKER y por tanto cada uno de los subproyectos entre los que se encuentra el Tracker lo utilizan.

Herramientas MySQL

MySQL Query Browser: es una herramienta gráfica proporcionada por MySQL AB para crear, ejecutar, y optimizar consultas en un ambiente gráfico, donde MySQL Administrator esta diseñado para administrar el servidor MySQL. MySQL Query Browser esta diseñado para proporcionar ayuda en las consultas y analizar datos almacenados en la base de datos MySQL.

Para la comprobación del funcionamiento del Tracker ha sido muy importante tener un entorno gráfico sobre el que comprobar de manera rápida y eficiente el estado de la BD,

ya que antes de realizar las consultas e inserciones desde el programa se puede saber su resultado ejecutándolas por separado mediante MySQL Query Browser. Esto ahorra muchos recursos (especialmente tiempo) al evitar la ejecución completa del programa de tracking.

3.3. Lucene

Se trata de una tecnología para la Recuperación de Información que realiza procesos de indización y búsqueda, cuenta con una API escrita en Java, también está disponible en otros lenguajes de programación, soporta la indización de documentos con formatos: txt, pdf, doc, ppt, rtf, xml y html.

Lucene es una novedosa herramienta que permite tanto la indización cómo la búsqueda de documentos. Creada bajo una metodología orientada a objetos e implementada completamente en Java, no se trata de una aplicación que se descarga, instala y ejecuta sino de una API flexible, a través de la cual se añaden, con esfuerzos de programación, capacidades de indización y búsqueda a cualquier sistema que se esté desarrollando. Existen otras herramientas, aparte de Lucene, que permiten realizar la indización y búsqueda de documentos pero dichas herramientas se utilizan para usos concretos, lo que implica que el intentar adaptarlas a un proyecto específico sea una tarea realmente difícil. La idea que engloba Lucene es completamente diferente, ya que su principal ventaja es su flexibilidad, permite su utilización en cualquier sistema que lleve a cabo procesos de indización o búsqueda. Lucene tiene versiones para otros lenguajes como Perl, C#, Ruby y C++. Para entrar más a detalle en el siguiente apartado se tratan los orígenes de Lucene.

El desarrollo y crecimiento masivo de las redes de computadoras y medios de almacenamiento a lo largo de los últimos años, motiva la aparición de un creciente interés por los sistemas de clasificación automática de documentos. Esta necesidad de búsqueda de datos en la web o en cualquier archivo que contenga texto dio origen a Lucene, para implementarse en cualquier aplicación o sistema que requiera un motor de búsqueda. Estos sistemas realizan diferentes operaciones de clasificación basándose en el análisis del contenido del texto de los documentos que procesan. La mayoría de las técnicas de análisis y representación de documentos utilizadas en la actualidad en los sistemas de clasificación, se basan en criterios fundamentalmente estadísticos, centrados en frecuencias de aparición de términos en los documentos.

Se distinguen dos fases en el proyecto basadas en la utilización de Lucene:

- En primer lugar indiza los documentos que contiene la BD Politiktracker.

- Después y a través de una consulta del usuario se encarga de buscar en el índice y mostrar los resultados con éxito.

A continuación se resumen las principales características que hacen de Lucene una herramienta muy útil para la realización del proyecto TRACKER:

- Es multiplataforma.
- Dispone de algoritmos de búsqueda fiables.
- Permite ordenar resultados por relevancia.
- Tiene lenguaje de consulta.
- Se puede adaptar para realizar *stemming* sobre las palabras de los textos y de esta manera realizar una búsqueda más eficiente.
- Permite la búsqueda y ordenación por cualquier campo.
- Posibilita la búsqueda mientras se actualiza el índice.

Lemur

Como sistema de Recuperación de Información, Lemur permite todas las etapas desde la indización a la búsqueda de documentos. Lemur aporta una poderosa API implementada en C++ y está diseñada para trabajar en todos los sistemas operativos, permite la indización incremental e indiza atributos de los documentos.

Xapian

Xapian es una biblioteca de funciones OpenSource de Recuperación de Información, para crear motores de búsqueda está escrita en C++, pero también se encuentra disponible en otros lenguajes como Perl, Python, PHP, Java, C#, and Ruby. Xapian contiene una API potente y adaptable que le facilita al programador los procesos de indización y búsqueda.

Terrier

El software de Terrier está escrito en Java y en código abierto, permitiendo realizar aportaciones por parte de los usuarios y contará con versiones en nueve lenguas, entre ellas la española. Facilita mucho el trabajo a los programadores ya que permite indizar una colección de documentos de forma que se sabe cuántos documentos contienen un término determinado.

Comparación de Lucene con Lémur, Terrier y Xapian

- Terrier y Lucene son implementados en Java, Lemur y Xapian en C++, aunque todas tienen soporte para otros lenguajes de programación.
- Todas indizan diferentes formatos de texto como: PDF, WORD, HTML, HTM, TXT, XML, RTF, entre otras.
- Lucene permite Stemming para varios idiomas, las demás tecnologías también.
- Lucene permite búsqueda mientras se actualiza el índice, lo que otras tecnologías no hacen.
- Lemur y Lucene permiten la indización incremental, Xapian y Terrier no.

- Todas trabajan con modelos probabilísticos, excepto Lucene que trabaja con el modelo de espacio vectorial.

- Todas las tecnologías son OpenSource (Software Libre).

- Lucene permite búsqueda por cualquier campo, las demás tecnologías no

Y sobre todas las anteriores destacan tres ventajas primordiales de Lucene:

  - La enorme documentación que posee.
  - Su gran base de usuarios, lo que permite su mayor desarrollo y mejoras.
  - Las herramientas de administración que tiene. Entre ellas Luke que ha sido utilizada para visualizar resultados de índices en el proyecto.

Teniendo en cuenta las anteriores ventajas, Lucene resultó perfecto para ser incorporado a la funcionalidad del TRACKER, realizando un papel primordial en la RI. Aunque hay varias tecnologías disponibles para la recuperación de información.

3.3.1. Luke

Luke es una herramienta de desarrollo y diagnóstico para índices Apache Lucene. Luke accede a los índices de Apache Lucene ya creados y permite mostrar y modificar los datos de diversas maneras:

- Navegar los documentos por número o por término.
- Ver documentos / copiar documentos al portapapeles.
- Ver la lista de los términos más frecuentes.
- Realizar búsquedas y navegar por los resultados.
- Analizar los resultados de una búsqueda.
- Eliminar documentos del índice.
- Editar los campos de un documento.
- Optimizar índices.
- Comprobar los tokens generados en un texto cualquiera usando distintos analizadores.

Al igual que MySQL Query Browser es muy útil para tratar con la base de datos del proyecto, Luke es absolutamente recomendable para tratar con los índices que genera Lucene. El entorno gráfico permite comprobar el funcionamiento del Tracker paso a paso puesto que se puede acceder a las noticias con las que trata,

Para el presente proyecto se ha elegido la última versión disponible de Luke, la 0.9.1.

* Una restricción que tiene Luke es que requiere una versión de Java 1.5 superior.

3.4. Eclipse

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent Azureus.
Eclipse es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Un ejemplo es el recientemente creado Eclipse Modeling Project, cubriendo casi todas las áreas de Model Driven Engineering.
Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

La versión actual de **Eclipse** dispone de las siguientes características:

- Editor de texto.
- Resaltado de sintaxis.
- Compilación en tiempo real.
- Pruebas unitarias con JUnit (conjunto de clases para ejecutar clases de Java de manera controlada).
- Control de versiones con CVS.
- Integración con Ant (herramienta para realizar tareas mecánicas y repetitivas).
- Asistentes (wizards) para la cración de proyectos, clases, test, etc.
- Refactorización.

Asimismo, a través de complementos libremente disponibles es posible añadir:

- Control de versiones con Subversión, SVN(sistema de control de versiones para sustituir a CVS).
- Integración con Hibernate (facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación).

Capítulo 4

Introducción al seguimiento de noticias

4.1. Definición y elementos de una noticia

Una noticia es el relato o redacción (cada cual con sus propias reglas de construcción enunciación) que refiere a un hecho novedoso o atípico -o la relación entre hechos novedosos y/o atípicos-, ocurrido dentro de una comunidad o determinado ámbito específico, que hace que merezca su divulgación.

La estructura de la noticia impresa puede tener alguno de estos elementos:

Titular

Es un texto muy breve, claro y preciso, que recoge una síntesis (generalmente en una o dos líneas) de lo que se informa posteriormente. Es el elemento más visible, debe referirse al contenido (por connotación o denotación).

En sus orígenes el texto noticioso ocupaba toda la página del periódico sin encabezamientos y los titulares se limitaban a servir de separación entre las diferentes noticias. Con la llegada del periodismo informativo (2ª mitad del siglo XIX) los periódicos empezaron a ordenar sus contenidos y a presentarlos de manera más atractiva, diferenciando las diversas noticias e introduciendo titulares más complejos y llamativos. Estos serían entonces el reclamo para atraer al público e interesar a los lectores.

El titular es muy importante, porque a veces es lo único que alcanzamos a leer y en muchas ocasiones es lo único que recordamos de una noticia, aunque la hayamos leído en su totalidad. Todo titular debe cumplir tres funciones: ser *atractivo* (llamar la atención del lector), *informativo* (dar cuenta del contenido de la noticia) y ser *objetivo* (exponer el contenido de la noticia) o *subjetivo* (exponer la opinión del autor o un aspecto segmentado de la noticia).

Elementos del titular

- *Cintillo*: sirve para vincular distintas informaciones que se relacionan temáticamente. Orienta al lector en la tarea de lectura. Ej.: Deportes, Internacional, Sociedad, Cultura...
- *Antetítulo*: precede al título y complementa aspectos informativos de la noticia que no aparecen en el titular. Se escribe en un cuerpo de letra menor que el título y con un tipo de letra diferente.
- *Título*: es el elemento fundamental del encabezamiento. Resume la entradilla o primer párrafo de la noticia.
- *Subtítulo*: amplia algunos detalles fundamentales apuntados en el título o en el antetítulo.
- *Ladillo*: es un pequeño título que se coloca dentro de la columna de texto y que aparece justificado a un lado. Se coloca para separar los distintos párrafos de la noticia. Suele ser bastante corto y no debe repetir palabras que se hayan utilizado antes en el título, antetítulo o subtítulo.
- *Sumario*: titulares que pretenden llamar la atención sobre aspectos del cuerpo de la noticia que no se incluyen en el encabezamiento. Son muy utilizados en revistas gráficas y de información general.

No obstante, algunas escuelas les dan nombres distintos a los elementos del titular:

- **Antetítulo**: Puede recibir el nombre de **Volanta**.
- **Subtítulo**: Recibe el nombre de **Copete** o **Bajada**.
- **Ladillo**: Pasa a llamarse **Subtítulo**.

Tipos de titulares. Lingüísticamente, se puede hablar de tres tipos de titulares:

- *Informativos*: identifican la acción y al protagonista.
- *Expresivos*: no persiguen íntegramente informar sobre un hecho, sino que intentan impactar a los lectores. Suelen ser de una palabra, aparecen en la primera página y son muy frecuentes en la prensa deportiva.
- *Apelativos*: utilizan el lenguaje para llamar la atención sobre un hecho del que no se informa en profundidad. Son propios de la prensa sensacionalista y de sucesos.

Por herencia, en función de su sistema jerárquico :

- *De intervalo abierto*: son los titulares en los cuales únicamente está presente el título.

- *D intervalo abierto a la derecha*: son aquellos titulares en los que solo aparece el título pero incluyen una especie de guía llamados señaladores deícticos (nos informan del tiempo, el espacio y la persona).
- *De intervalo abierto a la izquierda*: son aquellos titulares insertados dentro de la noticia que pretenden relajar la lectura y llamar la atención sobre un punto concreto de la noticia. Son conocidos también como intertítulos.
- *Con continuidad*: son aquellos titulares en los cuales se distingue claramente el título y el lead.
- *Con semicontinuidad inferior*: aquellos titulares que tienen subtítulo.
- *Con semicontinuidad superior*: aquellos titulares que tienen antetítulo.

Otras categorías:

- *Titulares temáticos*: mencionan genéricamente el tema sobre el que trata la noticia. Son titulares informativos, pero sólo tratan un elemento de la noticia sin aportar datos complementarios.
- *Titulares de actos de la palabra*: están basados en declaraciones (tanto orales como escritas)de personajes de actualidad. Pueden ser de tres tipos:
    - *Titulares con cita textual*: reproducen literalmente una declaración.
    - *Titulares en forma indirecta*: recogen las declaraciones sintetizándolas, reelaborándolas y condensándolas para transmitir la idea que tiene el protagonista.
    - *Titulares mixtos*: utilizan citas directas e indirectas. El periodista no utiliza la cita completa, pero sí algunas palabras puntuales.

Epígrafe

Suele estar ubicado en la página siguiente a la noticia y anterior al prólogo. Ofrece información sobre las fotografías y/o infografías y/o gráficos.

Bajada

La bajada es la que aclara el título y se encuentra dentro de la noticia, del cuerpo de ésta.

Cuerpo de la noticia

Se da la información completa. La información va de mayor a menor importancia.

Figura 4.1: Componentes básicos de un sistema de recuperación de información.

Sin embargo, por ser Internet un medio diferente del impreso, se hace necesario estructurar las noticias de forma diferente.

La extensión de la noticia

La extensión de la noticia y de todos los demás textos publicados en línea no puede ser demasiado larga, ya que los lectores en Internet rechazan instintivamente textos demasiado extensos.

Los artículos pensados para la edición en papel, cuando se trasladan sin adaptación al ordenador, resultan, por lo general, demasiado largos y obligan al lector a la engorrosa tarea de avanzar en el texto a lo largo de varias pantallas.

Leer en la pantalla no es lo mismo que leer en papel. La lectura en el computador cansa. Es por esta razón que la estructura de la noticia para la Internet debe ser de los más concisa, concreta y organizada posible, para permitirle al lector el llamado "escaneo", es decir, el conocimiento, a través de un vistazo, de los detalles sustanciales de la información.

La estructura de la noticia en la web

La correcta presentación de una noticia en Internet está relacionada con la técnica de la pirámide invertida, es decir, un tipo de redacción que organiza las ideas en una secuencia completamente opuesta a la utilizada tradicionalmente en los artículos científicos y académicos. Veamos:

Título:

Cada noticia consta del título que debe ser el resumen más exacto y fiel, pero además suficientemente llamativo del texto que sigue a continuación para atrapar la atención del usuario.

Encabezado (*lead, blurb*):

Es un párrafo inicial que guía al lector en el conocimiento del hecho y la puerta de entrada a la noticia. El encabezado contiene la esencia de la información, de manera que cualquier lector pueda tener la noción cabal de lo ocurrido con solo leer dicho párrafo. En este párrafo, no hay que tratar de incluir las respuestas a las seis preguntas: **¿qué?, ¿quién?, ¿cuándo?, ¿cómo?, ¿dónde? y ¿por qué?**, ya que es demasiada información para tan poco espacio. Según las reglas del periodismo tradicional, la respuesta a las seis preguntas debe estar incluida en los tres primeros párrafos del texto.

Cuerpo:

Constituye un desarrollo racional y coherente de lo propuesto en el encabezado, generalmente siguiendo el orden de importancia de los datos (la pirámide invertida): primero lo más relevante y luego lo de menos interés. Para facilitar a los usuarios la lectura en la pantalla, en las noticias largas es indispensable utilizar los subtítulos. Debido a que ningún lector posee la información completa acerca de todos los detalles de un acontecimiento, en caso de que alguna noticia esté relacionada con alguna otra, es necesario recordarle siempre al lector los datos básicos de lo ocurrido antes para darle continuidad al tema o utilizar los vínculos que constituyen una de las características principales y ventajosas de la escritura para la Web. De esta forma, podemos ampliar la información sin necesidad de ser repetitivos.

4.2. Modelos de noticias en la Base de Datos Politiktracker

Antes de almacenarse en la base de datos, las noticias están contenidas en formato RSS, que es una familia de formatos de fuentes web codificados en lenguaje XML. Un archivo RSS es un documento de texto compuesto por etiquetas acotadas, y se utiliza para suministrar a suscriptores de información actualizada frecuentemente. Estos archivos RSS serán descargados de las diferentes fuentes por el *Crawler* y almacenados en la BD.

Los archivos RSS, están estructurados en campos como los que siguen:

- **xml version**: Versión XML.
- **rss version**: Versión de RSS.
- **Canal** (channel) que está subordinado al campo rss con los metadatos del canal y del contenido de este.

Obligatorios:

- **title**: nombre del canal.
- **link**: La URL del servidor web del canal.
- **description**: La descripción del canal.

Opcionales:

- **language**: Idioma del canal.
- **copyright**: Copyright del canal.
- **managingEditor**: Nombre y e-mail del responsable del contenido.
- **webMaster**: Nombre y e-mail del responsable técnico del servidor.

- **pubDate**: Fecha de publicación.
- **category**: Taxonomia a la que pertenece el canal.
- **generator**: Software o aplicación utilizado para generar el canal.
- **docs**: URL que apunta a la documentación del formato RSS.
- **cloud**: Registro para ser notificado de las actualizaciones del canal.
- **ttl**: (Time To Live): minutos en que un canal puede permanecer en caché.
- **image**: imagen que se desea con el canal.
- **textInput**: Cuadro de diálogo para mostrarse con el canal.
- **skipHours**: Indicación para los agregadotes de las horas a evitar.
- **skipDays**: Indicación para los agregadotes de los días a evitar.

- **Item**: representa cada uno de los elementos o noticias de un canal. Contiene los siguientes campos (atención un máximo de 15 ítems por canal):

Obligatorios:
- **title**: Título del ítem
- **link**: URL o enlace del ítem
- **description**: Abstract o resumen del ítem

Opcionales:
- **author**: e-mail del autor del ítem.
- **category**: taxonomia o taxonomias a las que pertenece el ítem
- **comments**: URL de los comentarios relativos al ítem.
- **enclosure**: Descripción es caso de incluir objetos que forman parte del ítem (audio,..)
- **guid**: identificador unívoco del ítem.
- **pubDate**: Fecha de publicación
- **source**: El canal RSS o fuente del ítem

Pero no todos los campos anteriores son utilizados por el TRACKER.
Las noticias a descargar por la aplicación se obtienen de la tabla **post** de la base de datos. En esta tabla se encuentran los campos: Id_Post, Cod_Post, Prmalink, FechaHora, Titulo, Texto, Autor, Tendencia_Política_P, md5, Num_Comentarios, Num_A_Favor y Num_En_Contra.

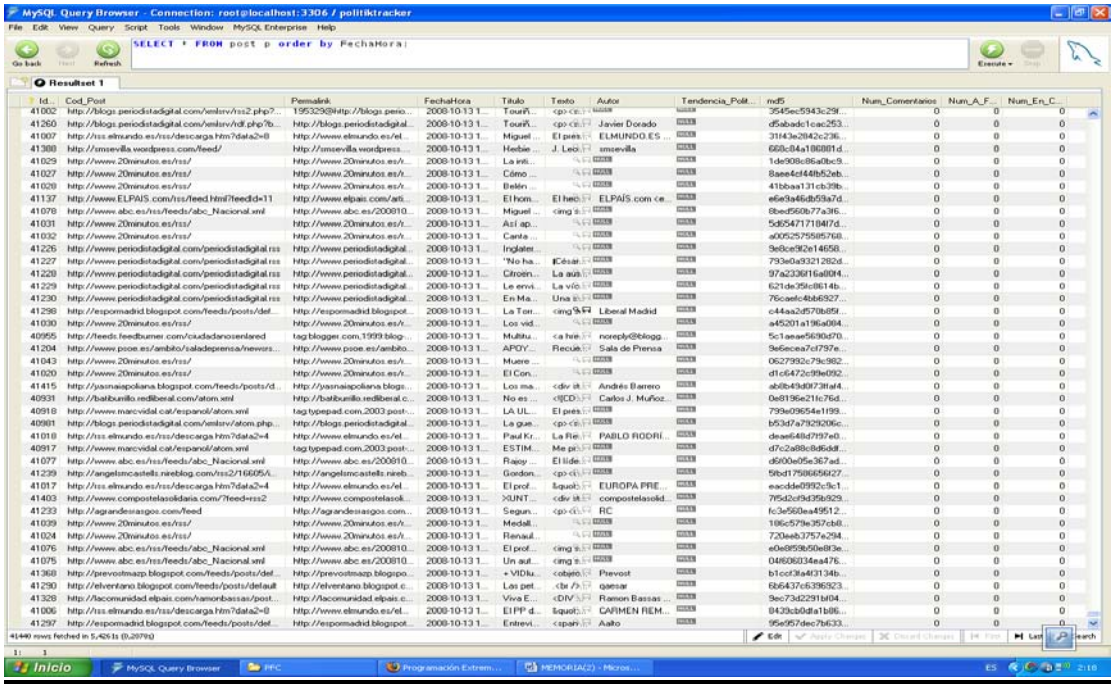


Figura 4.2: Campos de la tabla Post en la BD.

Aunque para el desarrollo de la aplicación TRACKER sólo se tienen en cuenta los campos: Id_Post, Titulo, Texto y FechaHora.

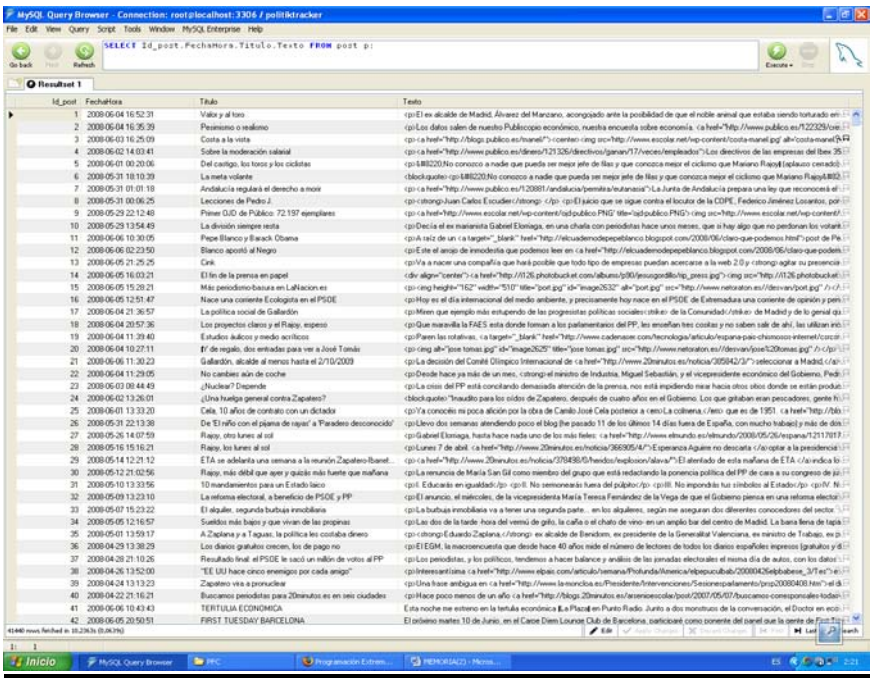


Figura 4.3: Campos de la tabla Post utilizados por el TRACKER.

Capítulo 5

Diseño del algoritmo de tracking

5.1. Planteamiento

Para la indexación y recuperación del contenido textual de los documentos que gestionamos nos bastaría con utilizar alguno de los analizadores que proporciona por defecto Lucene, pero si queremos potenciar las búsquedas de modo que no se produzca demasiado ruido en el resultado y para cumplir el objetivo de buscar documentos similares, tenemos que conseguir que los documentos pasen por un filtro lo más exhaustivo y restrictivo posible. Este filtro es el algoritmo que rige el funcionamiento del *tracker* para conseguir una buena relación de las noticias analizadas.

Un ejemplo de funcionamiento del algoritmo o AnalizadorCompleto que explica el funcionamiento de los filtros es el que sigue:

La cadena a tratar por el algoritmo es “La búsqueda de Elián González prosigue en EE.UU por parte del I.N.S. - Servicio de Inmigración y Naturalización”.

`STANDARD FILTER`

[La] [busqueda] [de] [Elian] [Gonzalez] [prosigue] [en] [EEUU] [por] [parte] [del] [INS] [Servicio] [de] [Inmigracion] [y] [Naturalizacion]

En este primer paso se han eliminado los signos de puntuación y los guiones.

LOWER CASE FILTER
[la] [busqueda] [de] [elian] [gonzalez] [prosigue] [en] [eeuu] [por] [parte] [del] [ins] [servicio] [de] [inmigracion] [y] [naturalizacion]

Con el segundo módulo del algoritmo se reducen todas las palabras a letras minúsculas.

STOP FILTER
[busqueda] [elian] [gonzalez] [prosigue] [eeuu] [parte] [ins] [servicio] [inmigracion] [naturalizacion]

Tras este paso se han eliminado las palabras irrelevantes sobre el sentido de la frase.

SPANISH STEM FILTER
[busc] [elian] [gonzalez] [proseg] [eeuu] [part] [ins] [serv] [inmigr] [natural]

Finalmente se reduce cada una de las palabras de la frase a su lexema.

Figura 5.1: Ejemplo de funcionamiento de *AnalizadorCompleto.*

Como se aprecia, la frase inicial ha quedado muy reducida lo cual permite realizar una búsqueda más eficiente.

Dicho algoritmo se construye en nuestra aplicación como una clase aislada de nombre **AnalizadorCompleto** que reúne las características necesarias para realizar un tracking eficiente. Para ello implementa los siguientes conceptos:

- **stopwords**: son una lista de palabras de uso frecuente que, tanto en la indexación como en la búsqueda, no se tienen en consideración, se omiten.
- **stemming**: es un método para obtener la raíz semántica de una palabra. Las palabras se reducen a su raíz o stem, de modo que, si buscamos por "abandonados" encontrará "abandonados" pero también "abandonadas", "abandonamos", esto es debido a que, en realidad, estamos buscando por la raíz de la palabra que es "abandon".
- **modelo de espacio vectorial**: es el modelo algebraico (expuesto en el punto 4.3.2.3. de esta memoria) utilizado para filtrar, indexar, recuperar y calcular la relevancia de la información. Representa los documentos con un lenguaje natural mediante el uso de vectores en un espacio lineal multidimensional. La

relevancia de un documento frente a una búsqueda puede calcularse usando la diferencia de ángulos de cada uno de los documentos respecto del vector de busca, utilizando el producto escalar entre el vector de búsqueda.

5.2. Elementos del algoritmo

El algoritmo de tracking de la aplicación tiene una estructura en la que se diferencian 4 grandes bloques o módulos que son los que van tratando y modificando el texto que recibe como entrada. Con las diferentes transformaciones se consiguen extraer del texto sus principales características excluyendo los elementos del mismo que sean irrelevantes.

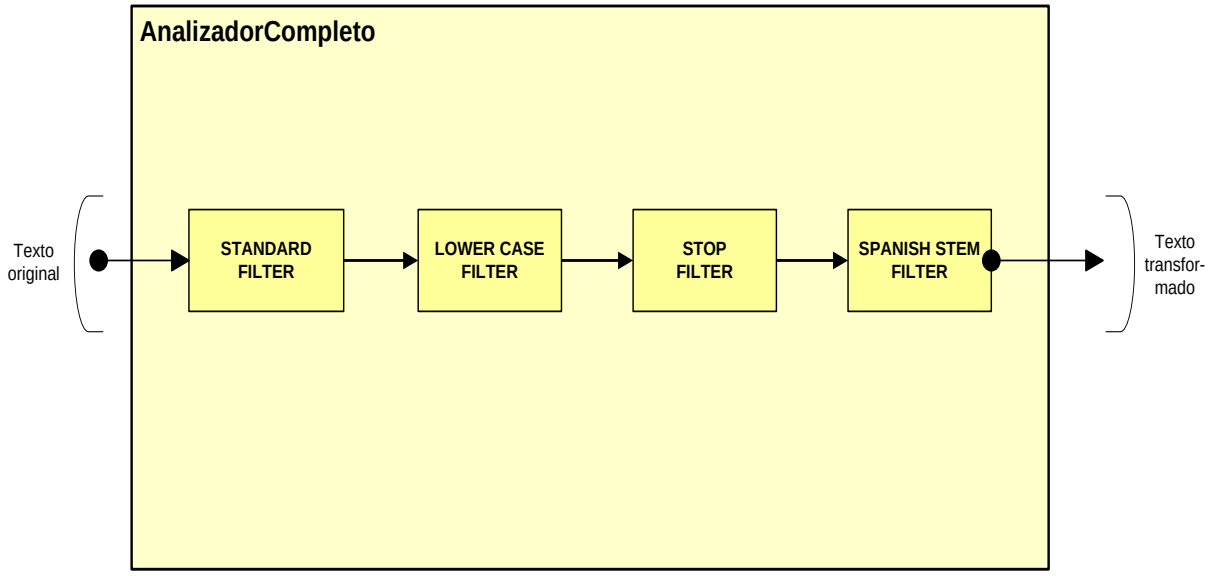


Figura 5.2: Filtros y transformaciones en AnalizadorCompleto.

El algoritmo de filtrado de documentos se utiliza tanto a la hora de la indexación de noticias como en el momento de realizar la búsqueda sobre los documentos. Al realizar la indexación es muy importante utilizar el analizador para tener un índice más reducido que contenga sólo la información relevante de cada noticia. Además, en el proceso de búsqueda de documentos sobre el índice se debe especificar la utilización del analizador para poder realizarlo de manera eficiente.

Las funciones y transformaciones que realizan los módulos del AnalizadorCompleto se detallan a continuación.

STANDARD FILTER

Normaliza cada una de las cadenas de texto que recibe utilizando un elemento de la librería Lucene denominado *StandardTokenizer*. Lo que hace este elemento es dividir el texto que recibe en tokens o palabras transformadas de la siguiente manera.

- Transforma las palabras según los signos de puntuación que contengan, eliminando los mismos de la palabra. Sin embargo, un punto que no esté seguido de un espacio en blanco es considerado parte del token o palabra transformada.
- Trata las palabras que tengan guiones en su interior eliminándolos, a no ser que haya un número en la palabra o token, en cuyo caso el token completo es interpretado por el tokenizador como un número y por tanto no se transforma.
- Reconoce direcciones de correo electrónico y de servidores de internet como un solo token.

La ventaja de este tokenizador y por tanto del *StandardFilter* es que puede ser utilizado para tratar textos de casi cualquier idioma europeo ya que los signos de puntuación, guiones y otros símbolos siguen unas normas comunes y su influencia sobre el significado de las palabras no resulta decisiva la mayor parte de las veces.

LOWER CASE FILTER

Normaliza el token o cadena que recibe convirtiéndolo a minúsculas. Es decir cambia todas las letras de la palabra que recibe a esa misma letra pero escrita en minúsculas.

Nota: este filtro produce un gran resultado para la mayoría de idiomas europeos, sin embargo tienen unas consecuencias nefastas para algunos idiomas asiáticos en los que las palabras no están separadas por espacios sino que las mayúsculas y minúsculas marcan el comienzo y fin de la palabra.

STOP FILTER

La función que realiza este filtro es la de eliminar las *stop words* o palabras rutinarias de la cadena que recibe. Al hablar de palabras rutinarias estamos hablando de palabras cuya frecuencia de aparición es muy alta y su influencia sobre el significado general de la frase es muy reducida. Ejemplos de este tipo de palabras serían las preposiciones, los artículos, algunos adverbios, etc.

La lista de palabras que elimina este analizador ha sido construida manualmente. Lo más idóneo sería que la aplicación tomase las *stopwords* de un fichero externo donde estuviese almacenada la lista. De esta manera se podría aumentar (o disminuir si se requiriese) la lista de palabras para poder realizar una mejor búsqueda de documentos, y sin necesidad de retocar el código de la aplicación. Esto se realiza con el consiguiente

coste adicional de tiempo, pero es un coste muy justificado teniendo en cuenta los beneficios.

Como es lógico, la lista de palabras *stopwords* es propia de cada idioma por lo que es indispensable crear una nueva lista de palabras si se van a tratar textos que contengan palabras en otros idiomas diferentes al español. En este sentido, palabras tomadas de otros idiomas al español como anglicismos o galicismos no se podrían eliminar a no ser que se realizase una lista de *stopwords* combinando varios idiomas, aspecto este que se ha desechado en la aplicación *tracker*.

SPANISH STEM FILTER

Este filtro utiliza la clase SpanishStemmer, de la librería lucene-snowball. Snowball es un lenguaje de programación para el manejo de strings que permite implementar fácilmente algoritmos de stemming.

Las técnicas de stemming o segmentación consisten en la reducción de las palabras a su raíz o lexema. Este lexema posee un significado autónomo e independiente y constituye la parte invariable de una palabra. (No es exactamente invariable, porque puede haber alomorfos; *poder* y *puede* ilustran este aspecto). Al constituir la parte esencial e invariable de una palabra, permite relacionar y hacer búsquedas en textos que contienen palabras referidas a lo mismo pero no escritas de la misma manera (p.ej. la arboleda referida al conjunto de árboles). Si no se aplicara este filtro, las relaciones entre palabras serían muy complicadas ya que requerirían una búsqueda al pie de la letra.

OTROS FILTROS

Hay muchos otros filtros que ofrece la librería Lucene que no se han implementado en el algoritmo de tracking. Ejemplos destacados de otros filtros podrían ser:

- El filtro de etiquetas HTML que no es necesario en la aplicación debido a que el texto que podría necesitar este filtro (campo texto de la base de datos) se trata con anterioridad mediante la clase *Limpia*.
- El filtro de sinónimos que permite tener en cuenta las palabras que se escriben de manera diferente y sin embargo tienen el mismo significado. Este filtro podría ser útil para corpus muy delimitados, pero para un grupo de datos tan amplio como el que se encuentra en *Politiktracker* no resulta eficiente implementar este filtro.

Antes de entrar en la estructura interna de los analizadores que componen el algoritmo de tracking del Analizador Completo, vamos a exponer cuál es el funcionamiento dentro de un *analizador*. Hay que resaltar que los analizadores intervienen en dos etapas del proceso: en la indexación y en la búsqueda.

Los términos que se extraen cuando se analiza el índice son los términos indexados. Esto, que puede parecer evidente, es una de las claves en la búsqueda de documentos ya que sólo los términos indexados pueden ser encontrados en la búsqueda. Si utilizamos el analizador completo para durante el indexado y durante la búsqueda hay que tener cuidado y saber exactamente lo que se guarda en el índice y lo que se espera recuperar de él.
En la fase de indexación se añaden documentos al índice, primero especificando la forma en que se guardarán los diferentes campos del documento:

```
private static void add(IndexWriter w, Post p) throws IOException {
            Document doc = new Document();
            doc.add(new Field("Id del post", p.getIdPost(), Field.Store.YES, Field.Index.TOKENIZED));
            doc.add(new Field("Titulo", p.getTitulo(),Field.Store.YES,Field.Index.TOKENIZED));
            doc.add(new Field("Texto", p.getTexto(),Field.Store.YES,Field.Index.TOKENIZED));
            doc.add(new Field("Fecha", p.getFecha(),Field.Store.YES,Field.Index.TOKENIZED));
            w.addDocument(doc);
        }
}
```

Esto es muy importante porque el Analizador convierte los datos para ser pasados al índice y luego se nutre de esos mismos datos para realizar las búsquedas. Por eso es necesario conocer todas las propiedades de los campos sobre los que pasan al índice.

Opciones para Field.Store:

- **Field.Store.YES**: Se guardará la información.
- **Field.Store.NO**: No se guardará la información.
- **Field.Store.Compress**: Se guardará pero de forma comprimida.

Opciones para Field.Index:

- **Field.Index.TOKENIZED**: Indexa el valor del campo para que pueda ser buscado. El analizador (en nuestro caso AnalizadorCompleto) será usado para tokenizar y posiblemente normalizar el texto antes de que sus términos sean almacenados en el índice. Esto es útil para texto común.
- **Field.Index.UNTOKENIZED**: Indexa el valor del campo sin usar el analizador para que pueda ser buscado. Como no se utiliza ningún analizador, el valor será almacenado como un solo término. Esto es útil para identificadores únicos como los Id_post.
- **Field.Index.NO**: El valor del campo no se indiza
- **Field.Index.ANALYZED_NO_NORMS**: Se analiza pero no se guarda la información sobre la longitud. Se indexa el valor del campo sin usar el analizador. Los beneficios de usar esta forma es que requiere menos

memoria ya que cada norma (norm) ocupa 1 byte. El campo norm se usa para calcular la relevancia del término en el texto.

Y por supuesto, especificando el analizador utilizado al indexar:

```
IndexWriter w = new IndexWriter(file,new AnalizadorCompleto(),true);
 add(w, p);
```

La clase AnalizadorCompleto es la clase básica del proceso y lo que hace es convertir el texto que recibe (en el caso del *tracker* son los diferentes campos de una noticia: titulo, texto, etc.) en una cadena de *tokens*. Estos tokens son palabras o palabras reducidas tras pasar por una transformación dentro del analizador.

Valoración de la similitud entre dos documentos

Para la valoración de los documentos, la aplicación recurre a la comparación de los campos título y texto de las noticias para que la comparación sea más completa. Algunas veces los títulos de noticias resultan ambigüos y otras es el texto de la noticia el que contiene mucha información confusa o directamente no contiene ningún tipo de información escrita (por ejemplo en noticias que sólo contienen videos o audios, el título resulta decisivo). Esta comparación se realiza de la siguiente en la aplicación manera:

IF *(SCORE* $_{TEXTO}$ *> UMBRALSIMILITUD*
OR *SCORE* $_{TITULO}$ *> UMBRALSIMILITUD)* ***THEN*** ***Noticias relacionadas***

Donde:

- *SCORE* $_{TEXTO}$ es la puntuación o grado de similitud que tienen las dos noticias o documento en lo que al campo texto se refiere
- *SCORE* $_{TITULO}$ es el grado de similitud del campo título de ambas noticias.
- *UMBRALSIMILITUD* es el factor de similitud a partir del cual se van a relacionar dos noticias. Este factor se fija manualmente y está directamente relacionado con el funcionamiento y efectividad del *tracker*, ya que si se pone un umbral demasiado bajo, la aplicación relacionará noticias que no son similares. Y si se pone un umbral demasiado alto, la aplicación no relacionará apenas noticias entre sí.

Hay que resaltar el uso de la cláusula **OR** en la comparación de noticias ya que la relación de dos noticias cualesquiera se realizará si sus campos título son parecidos **o** si sus campos texto lo son.

Se podría haber realizado la comparación de noticias de muchas maneras como por ejemplo realizando la media aritmética de la similitud por ambos campos. En este caso, dos noticias se relacionarían si esa media superara el umbral establecido.

Supongamos un ejemplo en el que dos noticias N1 y N2 tengan una similitud del 7 % por el campo texto y del 1 % por el campo título. Para un umbral de similitud establecido en la aplicación del 5 % significa que con nuestro método de comparación si se relacionarían ambas noticias pero con el método de la media de similitudes no (ya que la media de las similitudes por los dos campos sería del 4 %, menor que el umbral de similitud fijado).

Otra solución sería utilizar la media de ambas similitudes pero ponderándolas, por ejemplo dándole más importancias al campo texto que al campo título de las noticias. Esto no se ha realizado así en la aplicación ya que el corpus de noticias utilizado es bastante diverso y el formato de las noticias también lo es, por lo que se pierde eficacia si se resalta la importancia de alguno de los dos campos.

Puntuación de documentos en Lucene (Lucene scoring)

Después de haber utilizado el algoritmo de tracking para filtrar los documentos que se van a comparar para relacionar las noticias, es necesario tener un mecanismo de evaluación de la similitud entre los documentos Lucene.

En este apartado se explica la función utilizada por Lucene para la clasificación de documentos. Discutimos este apartado en este capítulo para que se tenga una idea general de los diferentes factores que intervienen en la puntuación que otorga Lucene a cada documento cuando se realizan búsquedas. Como en el *tracker* se realizan búsquedas sobre los documentos que representan noticias y que están almacenados en el índice, es muy importante saber cuál es método que tiene para relacionar noticias.

Para la clasificación de documentos el API de Lucene utiliza la función de puntuación o score que se muestra a continuación:

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \text{queryNorm}(q) \cdot \sum_{t \text{ en } q} \left(\text{tf}(t \text{ en } d) \cdot \text{idf}(t)^2 \cdot t.\text{getBoost}() \cdot \text{norm}(t,d) \right)$$

La puntuación de la búsqueda o query *q* está determinada por el producto vectorial o distancia-coseno que se define en el módelo de RI del Espacio Vectorial que utiliza Lucene y que se explica en el apartado 4.3.2.3. de la memoria. Un documento cuyo vector está muy cerca del vector de la query (o lo que es lo mismo, el ángulo entre los dos vectores es pequeño) tiene una alta puntuación de similitud.

1. **tf(t en d)** representa la frecuencia de cada término, definiendo frecuencia como el numero de veces que el término *t* aparece en el documento *d* que se está puntuando. Los documentos que tienen más ocurrencias de un término dado reciben una puntuación más alta.

El cálculo por defecto de *tf(t en d)* en la clase `DefaultSimilarity` es:

$$\texttt{tf(t en d)} = \text{frequency}^{½}$$

2. **idf(t)** significa frecuencia inverse del documento. Este valor se calcula con el inverso de *docFreq* (el número de documentos en los que aparece el término *t*). This means rarer terms give higher contribution to the total score. El cálculo por defecto de *idf(t)* en la clase `DefaultSimilarity` es:

$$\texttt{idf(t)} = 1 + \log\left(\text{numDocs} / \text{docFreq+1}\right)$$

3. **coord(q,d)** es un factor de puntuación basado en cuántos de los términos de búsqueda *q* se encuentran en el documento especificado *d*. Esto significa que un documento que contenga muchos de los términos de búsqueda, recibirá una puntuación para este factor más alta que otro documento que contenga menos términos de búsqueda.

4. **queryNorm(q)** es un factor de normalización utilizado para hacer las puntuaciones entre diferentes búsquedas comparables. Este factor no afecta al ranking del documento (debido a que todos los documentos clasificados se multiplicand por el mismo factor), pero intenta hacer las puntuaciones de diferentes querys (o incluso búsquedas de diferentes índices). El cálculo por defecto de este factor en la clase `DefaultSimilarity` es:

$$\text{queryNorm}(q) = \texttt{queryNorm(sumOfSquaredWeights)} = 1 / \text{sumOfSquaredWeights}^{1/2}$$

La suma de pesos al cuadrado (sumOfSquaredWeights) de los términos de búsqueda es calculada mediante la fórmula:

$$\texttt{sumOfSquaredWeights} = \texttt{q.getBoost()}^{2} \cdot \sum_{t \text{ en } q} \left(\text{idf}(t) \cdot \text{t.getBoost}() \right)^{2}$$

Donde:

5. **t.getBoost()** is a search time boost of term *t* in the query *q* as specified in the query text (see query syntax), or as set by application calls to `setBoost()`. Notice that there is really no direct API for accessing a boost of one term in a multi term query, but rather multi terms are represented in a query as multi `TermQuery` objects, and so the boost of a term in the query is accessible by calling the sub-query `getBoost()`.

6. **norm(t,d)** este factor encapsula (en tiempo de indexado) varios factores de mejora:
   - **Document boost** – se fija llamando a la función `doc.setBoost()` antes de añadir el documento al índice.
   - **Field boost** – es determinado por la función `field.setBoost()` antes de añadir el campo al documento.

- **`lengthNorm`**`(field)` – se calcula cuando se añade el documento al índice de acuerdo al número de tokens que contiene ese campo en el documento, de manera que campos cortos contribuyen más a la puntuación de este. LengthNorm se calcula por la clase Similarity mientras se está indexando.

Cuando se añade un documento al índice, los anteriores factores se multiplican. Si el documento tiene múltiples campos con el mismo nombre, todos sus factores de empuje o mejora se multiplican entre sí:

$$\text{norm(t,d)} = \texttt{doc.getBoost()} \cdot \texttt{lengthNorm(field)} \cdot \prod_{\substack{\text{campo } \boldsymbol{f} \text{ en} \\ d \text{ llamado } \boldsymbol{t}}} \texttt{f.getBoost()}$$

Sin embargo el valor resultante norm se codifica como un único byte antes de ser almacenado. Mientras se realiza la búsqueda, el byte de *norm* se lee del *index directory* y es decodificado para obtener un valor decimal de *norm* (en tipo float). Este proceso de codificación y decodificación, además de reducir el tamaño del índice, conlleva una pérdida en la precisión - no está asegurado que decodificar(codificar(x)) = x. Por ejemplo, el valor por defecto de realizar el proceso decode(encode(0.89)) no es 0.89 sino que es 0.75.

Como se ha explicado anteriormente, la mayoría de los factores de la fórmula de puntuación son controlados por una implementación de la clase *Similarity*. *DefaultSimilarity* es la implementación que se utiliza por defecto a no ser que se especifique otra diferente. Se pueden realizar modificaciones sobre la implementación de *DefaultSimilarity* para acomodar la puntuación a necesidades concretas. Por ejemplo, el factor de frecuencia de término **tf(t en d)** se puede modificar para que fuese un valor diferente de la raíz cuadrada de la frecuencia actual. En la práctica es muy raro necesitar un cambio en alguno de los factores de la fórmula de puntuación. Si se desea modificar alguno de los factores descritos, es recomendable acudir a la documentación Java de la clase Similarity y analizar profundamente el factor deseado.

Es importante resaltar que un cambio en los factores de mejora de tiempo de indexación o de los métodos de Similarity usados durante la indexación, requiere que el indice sea reconstruido de nuevo para que todos los factores estén sincronizados con él.

Capítulo 6

Análisis y diseño de la aplicación

6.1. La metodología utilizada: eXtreme Programming

La programación extrema o *eXtreme Programming* (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

¿Por qué XP?

En primer lugar se debe plantear la necesidad de utilizar una metodología. En el caso del proyecto fin de carrera como en el de cualquier proyecto informático es muy recomendable su utilización si se quieren ahorrar recursos (en especial tiempo) y trabajar de forma constante y dirigida hacia las metas u objetivos que se han definido para el proyecto.

A continuación se muestran las principales características de XP que me han guiado durante todas las fases de desarrollo de la aplicación:

- **Desarrollo iterativo e incremental**: pequeñas mejoras, unas tras otras.

- **Pruebas unitarias continuas**, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.

- Frecuente **integración del equipo de programación con el cliente** o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo. En el caso de la aplicación TRACKER las reuniones con el cliente (tutor de PFC) han sido constantes y básicas para el desarrollo.

- **Corrección de todos los errores** antes de añadir nueva funcionalidad. Hacer entregas frecuentes.

- **Refactorización del código**, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.

- **Simplicidad** en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Hay algunas formas de trabajo básicas de la metodología como es la programación en parejas que no se han podido aplicar debido a la especial condición de un proyecto de fin de carrera.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores.

Dadas las anteriores características anteriores se decidió optar por está metodología como guía de trabajo para el desarrollo del Tracker. Su **dinamismo** y **adaptación a las circunstancias** son sus principales ventajas a la hora de enfrentarse a un desarrollo bastante cambiante.

6.2. Estudio del entorno

Se plantea la necesidad de desarrollar una aplicación que permita el acceso a una base de datos para realizar un tratamiento de la información que contiene. Para ello se deben tener en cuenta las particularidades de la base de datos, el tratamiento que se quiere dar a la información y por supuesto el resultado que debe ofrecer el sistema Tracker.
El desarrollo de la aplicación está encuadrado en un proyecto mucho mayor denominado MEMETRACKER que busca realizar un análisis de las noticias políticas que se publican en los diferentes medios de comunicación en Internet (blogs, periódicos digitales, páginas relacionadas con la política, etc.). Las noticias, unidad principal de trabajo del proyecto, son descargadas de los diferentes sitios mediante un *Crawler* que posteriormente las inserta en el formato correspondiente en la base de datos. Es a partir del momento en el que se introducen nuevas noticias en la base de datos cuando entra en funcionamiento el Tracker que debe realizar una conexión segura a la misma y un acceso eficiente a los datos de noticias que se necesiten.
A pesar de alinearse dentro de un entorno muy concreto como es el del macroproyecto MEMETRACKER, se busca realizar una aplicación que sea adaptable a otros entornos y lo suficientemente genérica como para ser reutilizable bajo otras circunstancias como podría ser el cambio del origen de los datos. Para conseguir esto se requiere realizar el desarrollo modularmente de manera que se puedan modificar los módulos existentes de manera independiente y también agregar nuevos módulos en un futuro.
Al ser un proyecto novedoso dentro del sistema MEMETRACKER, no existe ninguna aplicación previa que realizase el proceso de *tracking*, por lo tanto es necesario crear un sistema completo partiendo de un pliego de requisitos y sin reutilizar ningún entorno existente.
Los usuarios del sistema son los propios investigadores del proyecto por lo que se les presupone las nociones de programación y bases de datos como para entender el funcionamiento de la aplicación. En cualquier caso la ejecución del proceso es totalmente automática y su puesta en funcionamiento muy sencilla.

6.3. Análisis de requisitos

A continuación se enumeran los requisitos por los que deberá regirse el sistema. Al tratarse de una aplicación experimental, los requisitos no provienen de ningún cliente sino del propio equipo de investigación (el desarrollador del sistema de *Crawler*, el diseñador y administrador de la base de datos, y por supuesto el diseñador del macroproyecto MEMETRACKER), por lo que el cumplimiento de los estándares de Ingeniería de Software para la especificación de requisitos se ha relajado, evitando incluir alguna información como la caracterización de requisitos. De esta forma, en el pliego de requisitos no se diferencia entre requisitos de usuario y requisitos de software, ya que el origen de ambos es el mismo y son de carácter combinado. Por lo tanto la única distinción que se realizará será entre requisitos funcionales (Rf) y requisitos no funcionales (Rn).

REQUISITOS FUNCIONALES

Rf – 01: El sistema TRACKER deberá tener un tiempo de ejecución total aceptable. 15 segundos o menos por cada 100 noticias relacionadas.

Rf – 02: Se relacionarán las noticias que se vayan descargando de la base de datos con las que hayan llegado en los últimos 30 días (con mayos diferencia temporal se considera que es complicado que dos noticias hablen del mismo tema).

Rf – 03: Teniendo en cuenta el requisito anterior, se deberán eliminar del índice utilizado los post con antigüedad mayor de 30 días con la fecha de ejecución. Esto se hará con el fin de onseguir mayor eficiencia en tiempo de ejecución y en espacio de almacenamiento.

Rf – 04: Cada vez que se ejecute la aplicación se debe abrir y cerrar la conexión con la BD Politiktracker utilizando los datos de conexión (usuario, contraseña, host, etc.) destinados a ello.

Rf – 05: En el proceso de relación de post formando grupos temáticos se modificarán exclusivamente las tablas noticia y agrupa de la BD.

Rf – 06: Cada post extraído de Politiktracker se relacionará únicamente con otro post de la BD, pudiendo relacionarse consigo mismo si no existe otro post parecido.

Rf – 07: Se creará un único índice que servirá como apoyo al proceso de relación de noticias. Este índice se creará en disco para permitir acceder a él y tener conocimiento de los post que tiene.

Rf – 08: Se debe filtrar la información contenida en la BD ya que algunos de los campos de las noticias contienen etiquetas HTML que interfieren en el proceso de relación de noticias.

Rf – 09: Los productos del sistema deben ser los archivos de los grupos creados por la aplicación, y los archivos con las relaciones entre todas noticias. Tanto los archivos de resultados como el índice utilizado se almacenarán en el mismo directorio.

REQUISITOS NO FUNCIONALES

Rn – 01: El sistema debe estar en capacidad de dar respuesta al acceso de todos los usuarios con tiempo de respuesta aceptable y uniforme.

Rn – 02: La aplicación debe estar disponible 100% o muy cercano a esta disponibilidad ya que las noticias con las que trabaja TRACKER provienen de la aplicación *Crawler* que trabaja 24 horas al día.

Rn – 03: El sistema debe ser construido sobre la base de un desarrollo evolutivo e incremental, de manera tal que nuevas funcionalidades y requerimientos relacionados puedan ser incorporados afectando el código existente de la menor manera posible; para ello deben incorporarse aspectos de reutilización de componentes.

Rn – 04: El sistema debe estar en capacidad de permitir en el futuro el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial.

Rn – 05: El sistema debe ser diseñado y construido con los mayores niveles de flexibilidad en cuanto a la parametrización de los tipos de datos.

Rn – 06: El sistema debe ser fácil de instalar en todas las plataformas de hardware y software, así como permitir su instalación en diferentes tamaños de configuraciones.

Rn – 07: Todo el sistema deberá estar complemente documentado. Cada uno de los componentes de software que forman parte de la solución propuesta deberán estar debidamente documentados tanto en el código fuente como en la memoria adjunta.

Rn – 08: El uso de la aplicación está restringido a usuarios expertos dentro del proyecto MEMETRACKER. Por esta razón no se autentican los usuarios del sistema.

Rn – 09: El sistema debe validar automáticamente los datos tratados en el transcurso del proceso de detección y seguimiento. En el proceso de validación de la información, se deben tener en cuenta aspectos tales como obligatoriedad de campos, longitud de caracteres permitida por campo, manejo de tipos de datos, etc.

Rn – 10: La solución deberá integrarse perfectamente dentro del proyecto general MEMETRACKER. En concreto, la aplicación debe interactuar con los módulos de *Crawler* y de Administración de la BD Politiktracker.

Rn – 11: Garantizar que la ejecución del sistema no afecte el desempeño de la base de datos.

6.4. Arquitectura del sistema

A continuación se muestra un esquema con la arquitectura general del sistema. El sistema esta compuesto de tres módulos principales (además del módulo general que gestiona todos los módulos) que son los que se encargan de realizar la tarea de detecccón y seguimiento de noticias.

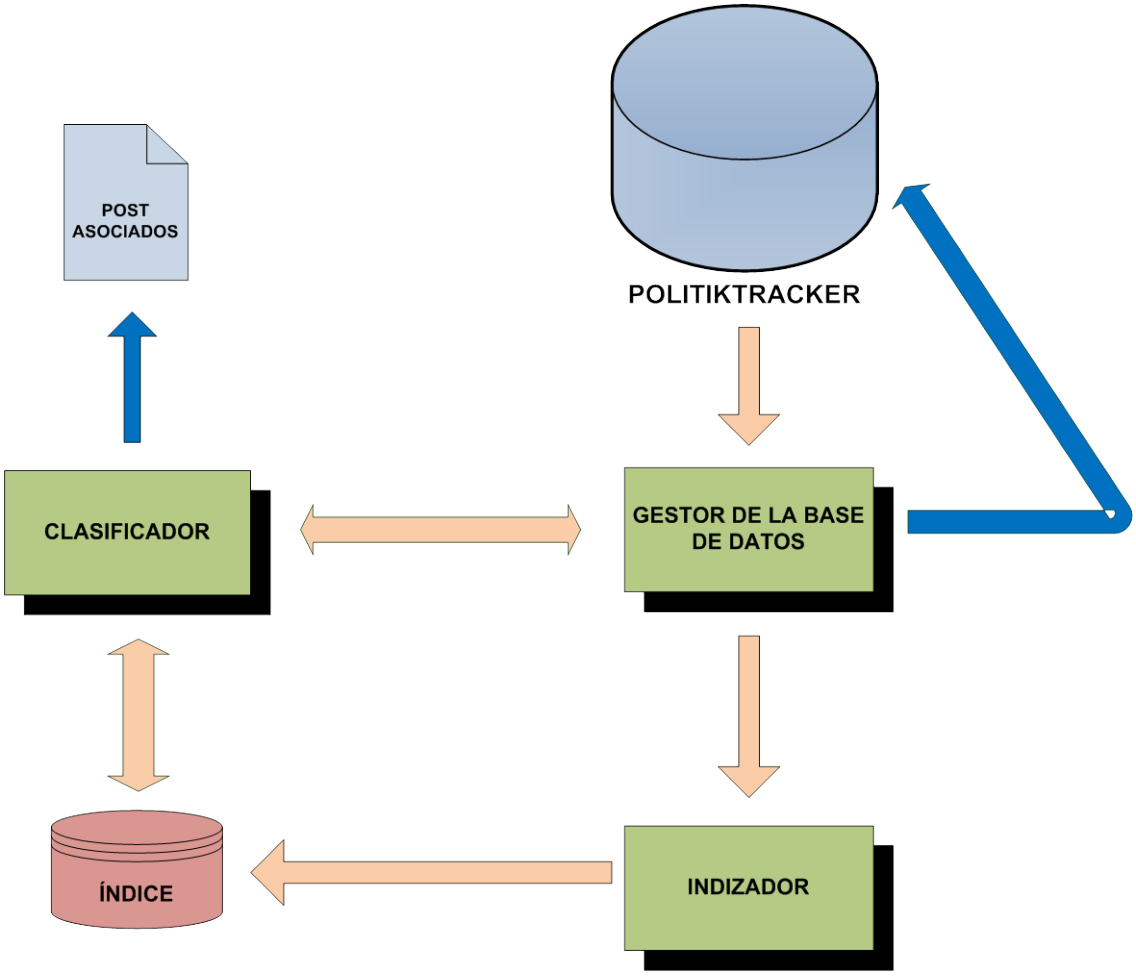


Figura 6.1: Arquitectura general del sistema.

El módulo Gestor de la Base de Datos se encargará de interactuar con la BD para extraer las noticias de ella y también para introducir los resultados finales en la misma.

Indizador es el módulo que gestiona el índice que permite el almacenamiento de las noticias y su búsqueda eficaz y eficiente.

El Clasificador interactúa con el índice para extraer la información necesaria de las noticias y poder relacionarlas entre sí.

6.5. Análisis previo

El objetivo de esta fase es el estudio de las necesidades de información que debe satisfacer el sistema TRACKER dentro del proyecto general MEMETRACKER, elaborando una serie de especificaciones formales que describan la funcionalidad del mismo y que permitan abordar con garantías la fase de diseño y sienten las bases del proyecto.

Partiendo del conjunto de requisitos establecidos anteriormente podemos realizar un primer análisis del sistema. Tal como se ve, el sistema deberá disponer de mecanismos para analizar las noticias contenidas en la base de datos. Deberá tratar el lenguaje HTML ya que las noticias que están guardadas en la base de datos contienen etiquetas HTML en algún campo. También deberá indexar las noticias para lo que utilizará el API que proporciona Lucene.

Se identifican tres módulos principales que el sistema deberá incluir, estos son: el módulo de interacción con la base de datos, el módulo de indización de noticias y el módulo de clasificación y seguimiento de noticias.

En primer lugar, el **módulo de interacción con la base de datos** *PolitikTracker* se encargará de interactuar con la misma abriendo la conexión para la descarga de los post que hayan llegado recientemente, y también insertará las relaciones que haya entre los post descargados en la tabla correspondiente de la base de datos cerrando la conexión cuando haya terminado sus trabajo.

Por otro lado tenemos el **módulo de indización** que procede a realizar un índice con todos los post que se han descargado de *PolitikTracker* para tener los datos almacenados de manera que luego se puedan relacionar las noticias de manera rápida y eficiente.

El último módulo principal de la aplicación de tracking es el **módulo clasificador** que recurre a la lista de noticias descargadas de la base de datos y las compara una a una con las noticias almacenadas en los índices. De esta manera podrá determinar si existe alguna noticia relacionada o si por el contrario es la primera noticia sobre un tema.
A continuación se muestran unos diagramas iníciales:

Diagrama de clases inicial:

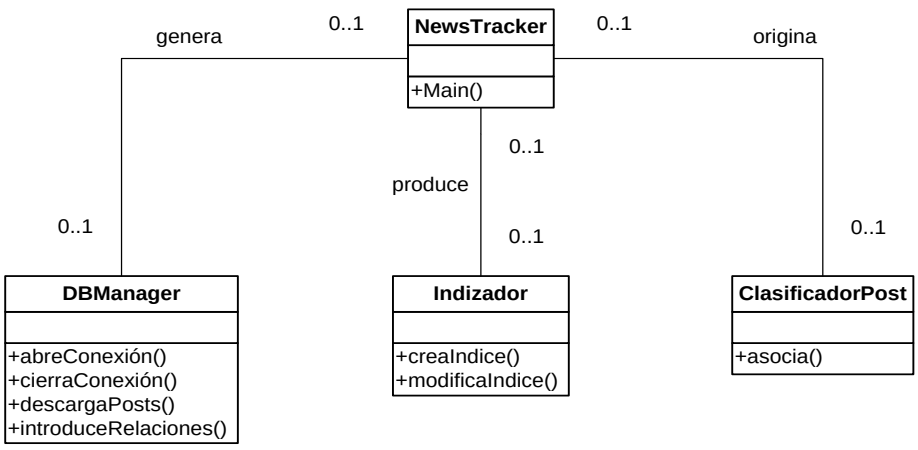


Figura 6.2: Diagrama de clases inicial.

Este es el funcionamiento básico del sistema inicial. Tenemos una clase *NewsTracker* que es la que rige el sistema y regula el uso de las otras tres clases, decidiendo en cada momento cuando se trata con la base datos, con el índice o con el clasificador de noticias.

Diagrama de objetos inicial:

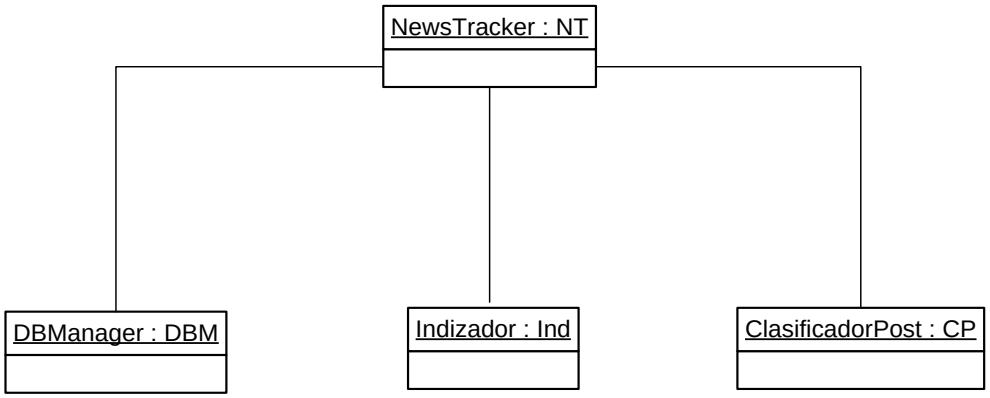


Figura 6.3: Diagrama de objetos inicial.

El sistema dispone de un objeto **DBManager** en cuyo estado inicial aún no ha creado la lista de post descargados. El objeto **Indizador** tampoco ha creado el índice en el disco duro (también sería posible crear el objeto índice en memoria, pero en nuestro caso se crea en disco).

Diagrama de objetos intermedio:

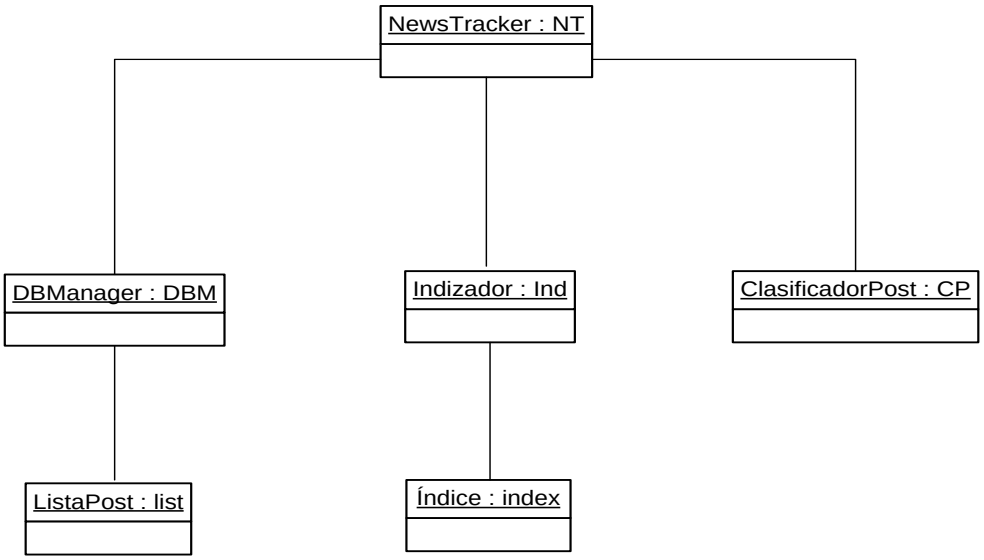


Figura 6.4: Diagrama de objetos intermedio.

Una vez que el TRACKER ha empezado su ejecución, el gestor de la BD crea la lista de post descargados. A partir de esa lista de post descargados se crea el índice con el que se van a relacionar las noticias. Es por esto que en el anterior diagrama de objetos se muestran dos objetos nuevos como son *list* e *index*.

6.6. Solución

La solución implementada para este problema es un *tracker* de noticias de una base de datos. Por tanto, su funcionalidad principal es la de realizar un seguimiento de las noticias contenidas en una base de datos. Este seguimiento consiste en el caso del presente proyecto en la relación de los contenidos que se extraen, pero esa relación debe verse plasmada de alguna manera.

Cuando la aplicación se ejecuta, las noticias son relacionadas por el sistema dando lugar a tres productos o resultados:

- Archivo de texto con las relaciones y su peso.
- Archivo de tipo XML con los diferentes grupos de noticias determinados.
- Modificación de la base de datos, en concreto de las tablas **agrupa** y **noticia.**

Creación de post_asociados.txt

Este es el archivo donde se muestra la relación de las noticias en forma de lista de parejas de noticias. En cada línea del archivo de texto se contiene un Id de post y a su lado el Id del post con el que el TRACKER lo ha relacionado. Además de esto también se muestra el grado de similitud con el que se han relacionado ambos post o noticias.

ID POST	ID POST SIMILAR	SIMILITUD ENTRE AMBOS POSTS

Así pues el archivo contendrá tantas líneas como relaciones entre noticias haya encontrado la aplicación.

Un ejemplo de línea escrita en post_asociados.txt podría ser la siguiente:

```
41189 --> 41425 : 0.67137235
```

Donde el post con Id 41189 ha sido relacionado por el TRACKER con el post de Id 41425, y se ha encontrado un coeficiente de similitud de 0,67137235, los cual equivaldría a que la aplicación a determinado que las dos noticias se parecen aproximadamente en un 67%.

Creación de grupos_post.xml

En este archivo se encuentran almacenados los grupos de noticias que ha producido el TRACKER. Los grupos se almacenan en un archivo *.xml* ya que posteriormente será útil tenerlo en este formato para la evaluación de la aplicación.

La forma en que se guardan los grupos en el archivo es mediante entidades que contienen identificadores de posts. Cada entidad es un tema o grupo que contiene los Id de las noticias que están asociadas a ese grupo.

No se almacena ninguna información diferente de los Id de cada noticia porque no se necesita más información. El proceso de relación ya ha terminado y el texto y título de cada post no se necesitará para nada más.

```
<entity id="1" name="" notes="">
<doc rank="41089" notes="" />
<doc rank="41200" notes="" />
<doc rank="41192" notes="" />
<doc rank="41197" notes="" />
</entity>
<entity id="2" name="" notes="">
<doc rank="41174" notes="" />
<doc rank="41431" notes="" />
<doc rank="41367" notes="" />
<doc rank="41432" notes="" />
</entity>
```

Figura 6.5: Ejemplo de grupos creados por la aplicación.

Modificación de la base de datos

Además de la creación del archivo de post asociados, las relaciones entre las diferentes noticias deben almacenarse en la base de datos *Politiktracker*. Este era el objetivo principal del TRACKER ya que los datos con los que trabajaba provenían de la base de datos. Los usuarios que accedan a la BD para consultar las noticias deben conocer las relaciones de las mismas sin necesidad de ejecutar la aplicación y acceder al archivo de texto de asociados.

La base de datos del macroproyecto MEMETRACKER está compuesta de casi 30 tablas. En el caso de esta aplicación sólo se modifican 2 tablas que son **agrupa** y **noticia.** El resto de tablas son modificadas por otras aplicaciones (como el *Crawler* o la aplicación de popularidad) o bien están creadas para permitir dar cabida a otra aplicaciones en el futuro.

Además de las tablas que modifica el TRACKER hay que reseñar que la aplicación toma los datos de la tabla **post**.

A continuación se presenta el modelo entidad-relación de la base de datos donde se muestran las relaciones de las tablas.

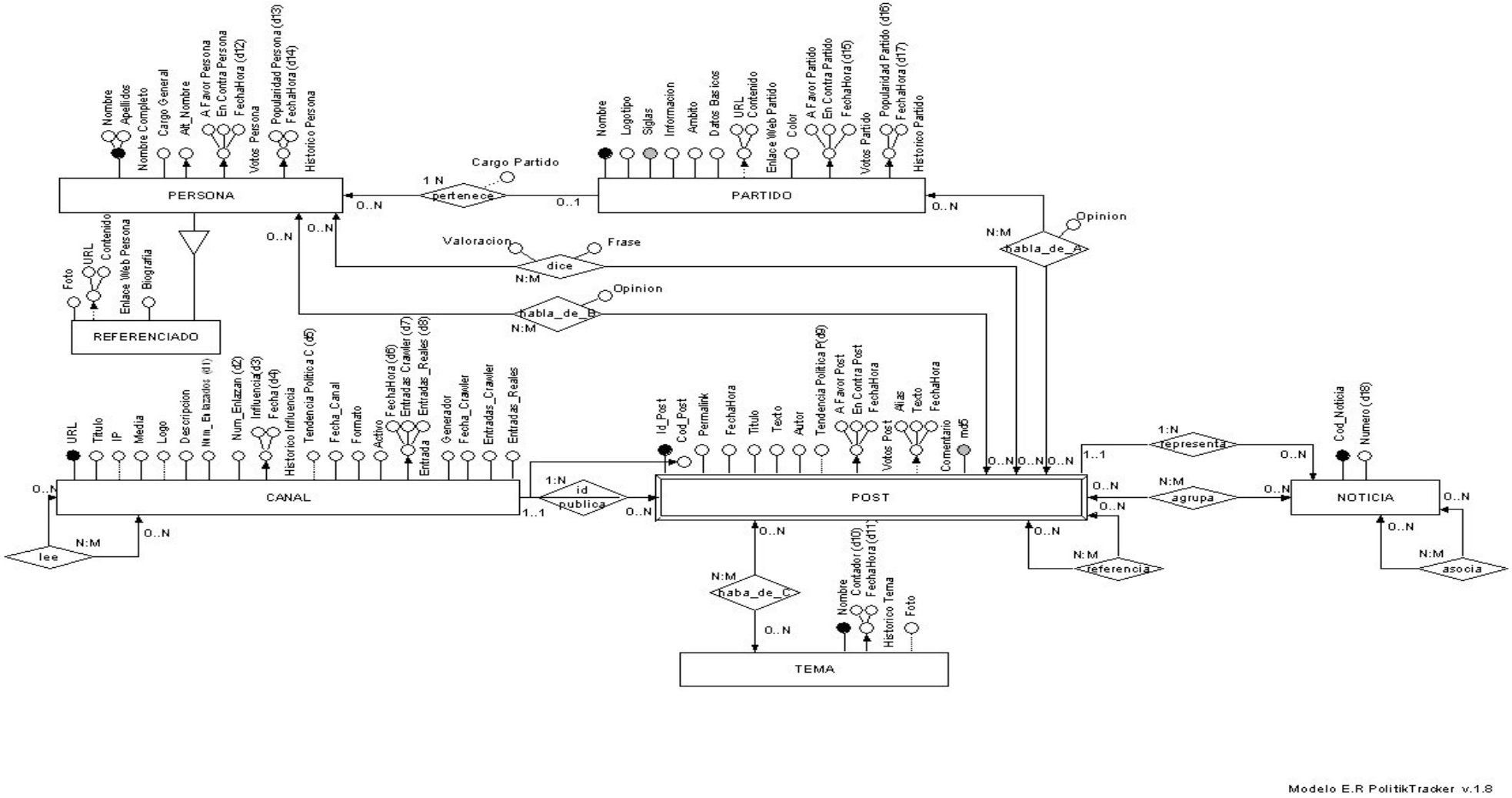


Figura 6.6: Modelo entidad – relación de la base de datos.

De la base de datos a la aplicación de tracking sólo nos interesa la relación que hay entre los POST y NOTICIAS.

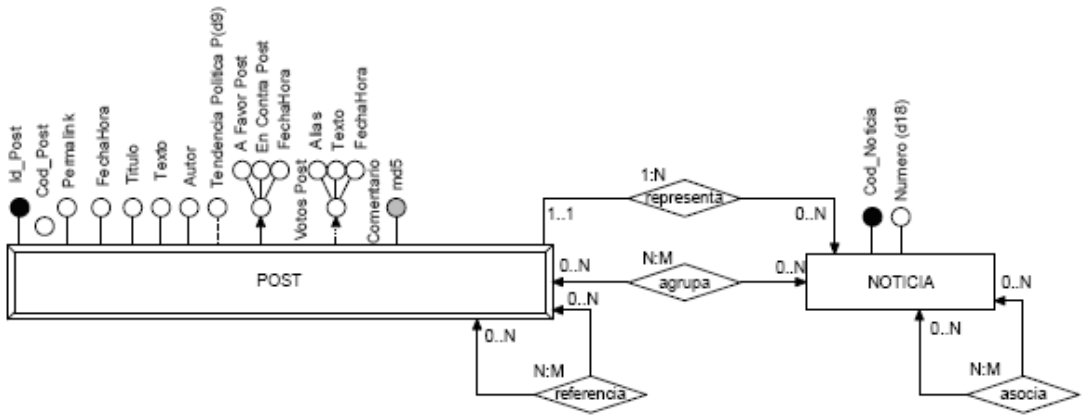


Figura 6.7: Detalle de las tablas Post y Noticia del diagrama ER.

DICCIONARIO DE TÉRMINOS DEL DIAGRAMA ENTIDAD-RELACIÓN

POST

Id_Post: Identificador numérico que lista las filas.
Cod_Post: URL del canal al que pertenece.
Permalink: Enlace permanente al post.
FechaHora: Fecha y Hora en la que fue publicado el post.
Titulo: Titulo del post.
Texto: Texto que resume la noticia.
Autor: Autor de dicho post.
***Tendencia Política P* (d9)**: Atributo derivado que define la posición política que defiende dicho post. Este atributo se extraerá automáticamente del texto.
Votos Post: Atributo compuesto que refleja un histórico de los votos que los usuarios del sistema han dado a un post en una fecha concreta.
A Favor Post: Indica el número de votos positivos para el post.
En Contra Post: Indica el número de votos negativos para el post.
FechaHora: Guarda la fecha y la hora.
Comentarios: Atributo compuesto que guarda los comentarios de los usuarios del sistema respecto a un post en una fecha concreta.
Alias: Nombre del usuario que hace el comentario
Texto: Refleja el comentario vertido por el usuario.
FechaHora: Indica la fecha y la hora en la que se hizo el comentario.
md5: Resultado de calcular el algoritmo con los valores “Titulo” y “Permalink”. Este valor será único para cada fila.

NOTICIA

Cod_Noticia: Atributo con el que identificamos todas las noticias del sistema.
***Número* (d18):** Atributo derivado que refleja el número de post que forman parte de la noticia. Se calcula sumando todos los post de la noticia.
A continuación se realiza la transformación (de la parte que afecta a la aplicación TRACKER) del modelo entidad-relación al esquema relacional.

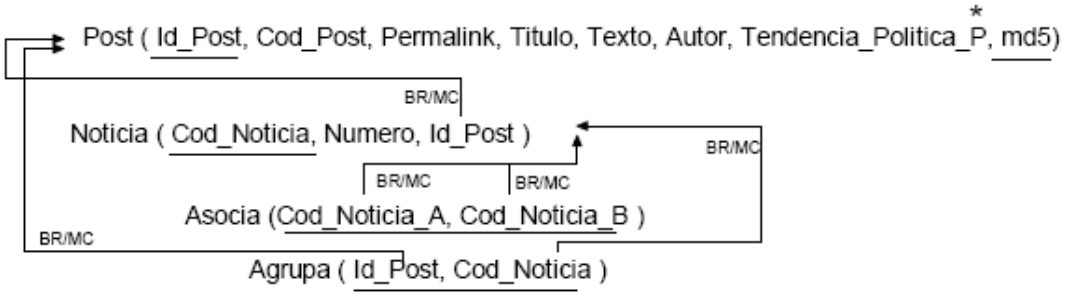


Figura 6.8: Diagrama relacional de tablas asociadas al TRACKER.

En este esquema en primer lugar se distingue la relación AGRUPA que surge de la unión entre “POST” y “NOTICIA”. En una “NOTICIA” quedan agrupados muchos “POST”, y un POST puede estar agrupado en muchas “NOTICIAS”, por tanto tenemos una interrelación N:M.

La interrelación reflexiva “*asocia*" se representa con la relación “ASOCIA”, que toma como clave principal las claves principales pertenecientes a las noticias que quedan asociadas entre si. Las opciones de borrado serán restringidas ya que no queremos que si una noticia se borra, también queden eliminadas las noticias que tenía asociadas.

En la tabla **post** se encuentran todas las noticias que debe descargar el TRACKER y ser analizadas para poder relacionarse con otras noticias. En esta tabla se guarda información de muchos atributos de los cuales sólo resultan interesantes para el programa aquellos que se almacenan en el índice, como son el Id_Post, el Título, el Texto y la Fecha de descarga del post. Aunque es posible que atributos como el Permalink u origen de la noticia, o su autor sean también utilizados por la aplicación.

Como resultado de la ejecución del TRACKER, en la interrelación **agrupa** se insertan datos de los post relacionados con otros post. Es decir se insertan los datos de cada una de las filas de *post_asociados.txt* a excepción de los datos referidos al coeficiente de similitud entre los post. Es decir, en esta tabla se realizan inserciones del Id_Post y Cod_Noticia.

Id_Post	Cod_Noticia

Donde Cod_Noticia es la noticia original o antigua con la que el sistema relaciona el post nuevo identificado por Id_Post.

Además de la tabla agrupa, también se modifica la tabla **noticia.** En esta tabla se introducen los datos de las noticias más antiguas sobre un tema. Estas noticias servirán de referencia a los post nuevos que vayan llegando a la base de datos y que hablen también sobre el tema de la noticia.

En la tabla noticia se guardan los campos Cod_Noticia, Número, Id_Post y FechaHora, que ofrecerán la información necesaria sobre la noticia y el tema referido.

Cod_Noticia	Número	Id_Post	FechaHora

Cod_Noticia es el mismo Cod_Noticia de la tabla agrupa y es el Id del post de la noticia original sobre un tema.
Número es un atributo que sirve para determinar el número de noticias relacionadas con el tema.
Id_Post coincide con el valor de Cod_Noticia ya que es el valor del Id de la noticia.
FechaHora refleja la fecha y la hora a las que el TRACKER introdujo la noticia en la base de datos *Politiktracker*.

6.7. Fase de análisis

En esta primera fase se tiene como objetivo modelar el problema de forma universal mediante orientación a objetos, por ello, se ha utilizado una concepción basada en clases y objetos y un diagrama de clases de análisis expresado mediante **UML**. Las clases identificadas son las siguientes:

- **NewsTracker**: es la clase principal del sistema. Se encarga de hacer de nexo de unión entre el resto de clases y de gestionar la ejecución del sistema.

- **DBManager**: es la clase encargada de interactuar con la base de datos *PolitikTracker*. Recibe la llamada de la clase ***NewsTracker*** para abrir y cerrar la conexión con la base de datos, de descargar los post que hayan llegado recientemente , y también de insertar las tuplas de noticias relacionadas en la tabla correspondiente de la base de datos.

- **Indizador**: esta clase es la encargada de crear el índice que sirve para relacionar noticias. Recibe la llamada de la clase ***NewsTracker*** y se encargará de añadir documentos al índice (cada noticia nueva que llega a la base de datos significa un nuevo documento en el índice), y de eliminar del mismo los documentos que sean antiguos.

- **ClasificaPost:** es llamada también por la clase ***NewsTracker*** y la función asignada a esta clase consiste en comparar todos los objetos de tipo post que recibe con los post que se encuentran almacenados en forma de documento en el índice. Una vez hecha la comparación, asocia un post con otro post similar que hubiera en el índice (si lo hubiera) según fuesen similares su campo título o su campo texto.

- **Limpia**: cuando la clase ***DBManager*** llama a esta clase es cuando realiza su cometido más importante que es limpiar los datos de los post descargados de la base de datos de las etiquetas HTML que contengan. Su otra función es la del borrado del índice una vez que se hayan relacionado las noticias por el campo texto para poder relacionarlas después por el campo post.

- **AnalizadorCompleto:** esta es una de las clases importantes de la aplicación puesto que contiene elementos y métodos que se utilizan tanto en el proceso de creación del índice como en el de búsqueda sobre índice. Lo utilizan las clases ***ClasificaPost*** e ***Indizador*** para tratar el texto que proviene de la BD de manera que se eliminan letras mayúsculas, palabras muy frecuentes y poco útiles para comparar, o también para dejar todas las palabras reducidas a su raíz semántica y mejorar los resultados. Al realizar búsquedas sobre el índice filtra la información contenida en éste de la misma forma.

Las siguientes cuatro clases son solamente clases estructurales y su única función es la de contener los datos de las noticias de la base de datos.

- **Noticia:** es la clase que guardará los datos de las noticias que hayan sido relacionadas por la clase *ClasificaPost*. Sólo interesará que guarden los valores de su identificador, del Id del post con el que se relaciona y de la fecha y la hora en la que se han relacionado por el programa *tracker*.

- **NoticiaNueva:** es la clase hija de la clase **Noticia** y su única función es la de almacenar los datos de una noticia nueva para su posterior introducción en la base de datos.

- **Post:** es la clase que sirve para guardar los datos que son importantes para el *tracker* de la base datos. Estos datos son el Id de cada post, su título, su texto y su fecha de almacenamiento en *Politiktracker*.

- **PostSimilar:** la clase hija de **Post** extiende la funcionalidad de la clase padre para guardar también el Id del post similar al que se está tratando.

Como se puede observar, en la fase de análisis las clases identificadas son totalmente genéricas, independientes del lenguaje de programación elegido. En el diagrama que se muestra a continuación se puede ver de forma grafica dicho análisis.

Diagrama de clases de análisis:

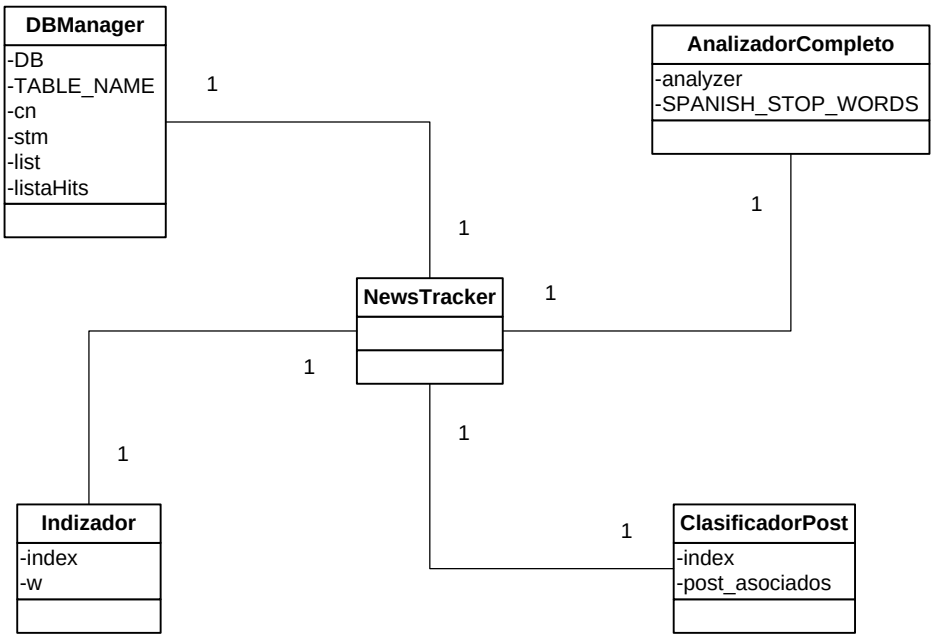


Figura 6.9: Diagrama de clases de análisis.

6.8. Fase de diseño

El diseño de la aplicación se realiza sobre la estructura marcada en el esquema siguiente donde se pueden apreciar las diferentes acciones principales y su interacción.

Estas acciones principales darán lugar a las clases principales del programa y conforman la funcionalidad completa de la aplicación que se ve representada por **NewsTracker**, la clase principal que englobará al resto de clases existentes.

Como se puede apreciar, no se muestran las clases auxiliares que sólo están diseñadas para servir de soporte para las otras clases. Estas clases son las que tienen que ver con las unidades de datos descargadas de la BD, las noticias. Las clases auxiliares se utilizan a lo largo de todo el proceso de tracking por el resto de las clases de una manera o de otra.

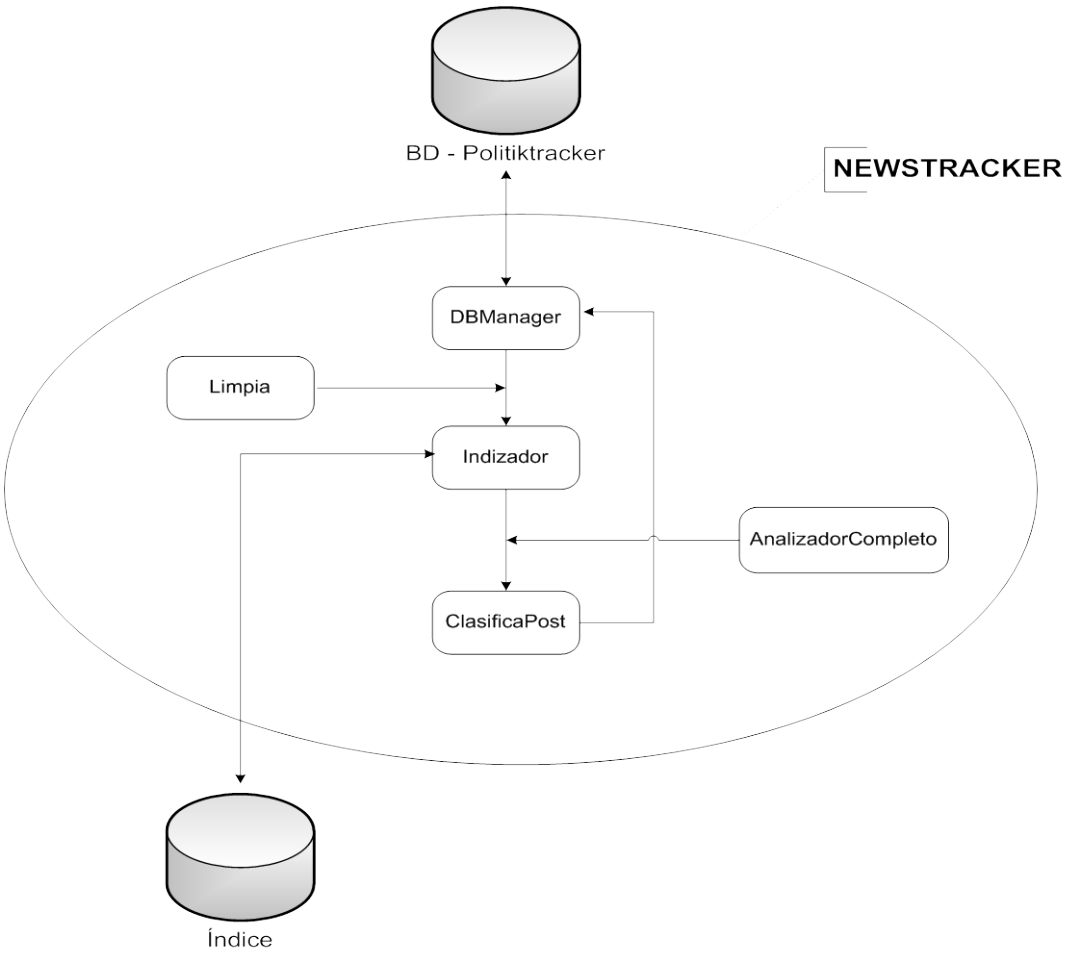


Figura 6.10: Esquema de diseño de la aplicación.

Una vez realizado un análisis del sistema e identificado las clases principales el siguiente paso es realizar un diseño adaptado a la implementación. Se ha escogido el lenguaje **JAVA** para implementar el sistema, utilizando **Eclipse** como entorno de desarrollo. Esta elección se basa en la necesidad de utilizar un entorno robusto con un lenguaje orientado a objetos de uso general y libre.
Una vez realizados los pasos previos, se deben concretar las clases que vamos a utilizar. Para realizar este proceso se diseñó el sistema de forma que cada clase tuviera una responsabilidad concreta y aportara una funcionalidad al sistema. El producto de este trabajo se puede ver en el diagrama de clases expuesto a continuación.

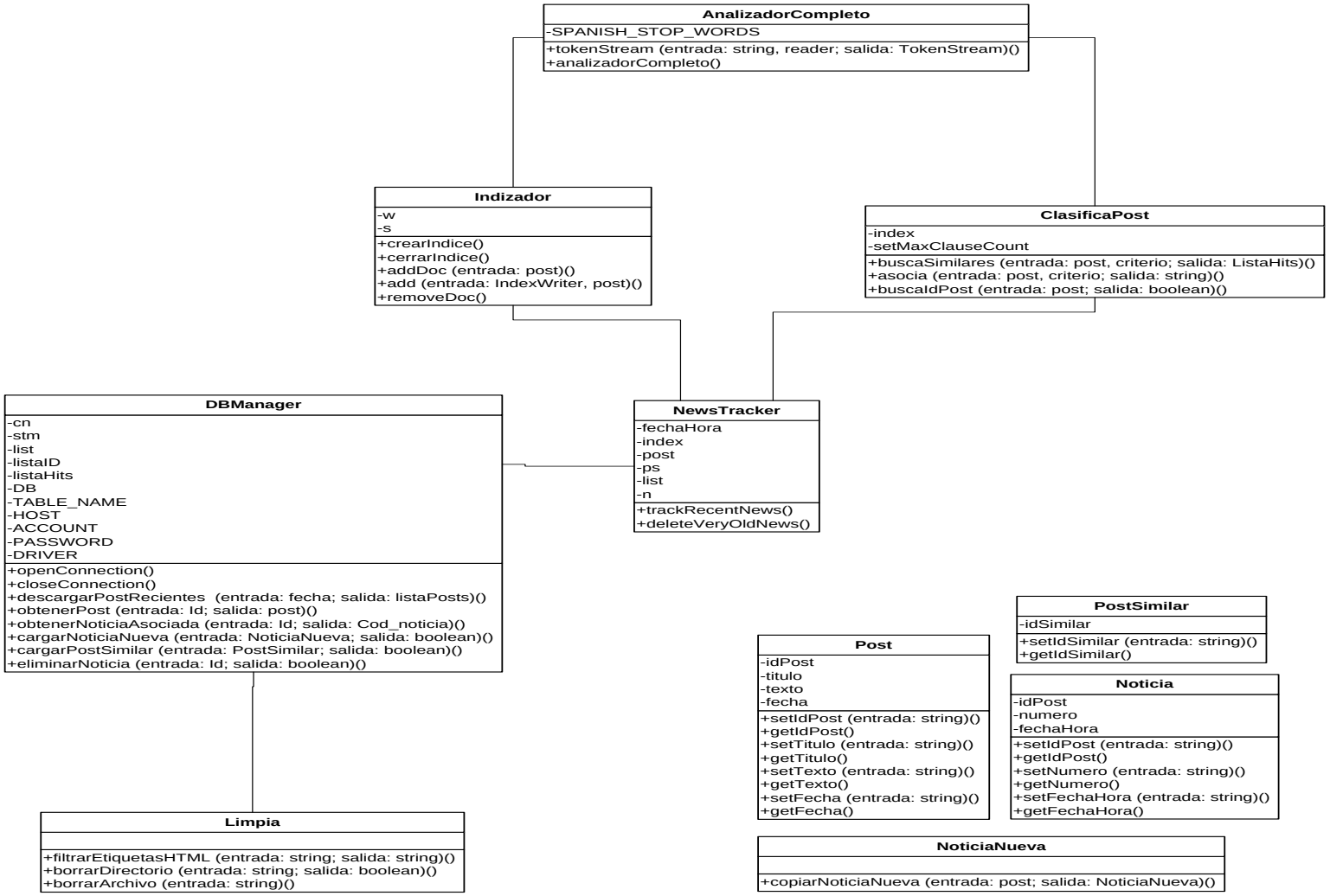


Figura 6.11: Diagrama de clases general de diseño.

Como se aprecia en el diagrama de clases, disponemos de seis clases principales. Cada clase dispone de una determinada tarea dentro del sistema. Para tener una definición más detallada de las mismas vamos a hacer una "ficha" de cada una, describiendo los métodos de que dispone.

Nombre: *NewsTracker*	**Tipo:** Concreta
Superclase:	
Subclase:	
Responsabilidades:	**Colaboraciones:**
Gestión del sistema	
Creación de los objetos principales de listas de post y coincidencias	
Muestra de los resultados	

Atributos y variables principales:

- **fechaHora:** fecha y hora en la que se empieza a ejecutar el TRACKER.
- **index:** ruta del directorio en el que se encuentra el índice.
- **post:** contenedor de los datos de cada post descargado.
- **ps:** contenedor de los datos de cada post similar.
- **list:** lista de post similares a un post determinado.
- **n:** contenedor de los datos de una noticia que se guardará en la base de datos.

Métodos principales:

- **trackRecentNews:** método general de la aplicación por el cual se pone en marcha el TRACKER.
- **deleteVeryOldNews:** método utilizado para eliminar del índice los documentos (noticias) que tengan una antigüedad mayor de 1 mes.

Nombre: *DBManager*	**Tipo:** Concreta
Superclase:	
Subclase:	
Responsabilidades: Conectar y desconectar con la base de datos *Politiktracker*	**Colaboraciones:**
Descargar los post que hayan llegado recientemente a la BD	
Realizar consultas, borrado y sobre todo inserciones de las relaciones entre las noticias en la BD	

Atributos y variables principales:

- **cn:** objeto de tipo *connection* que permite realizar la conexión con la BD mediante JDBC.
- **stm:** objeto de tipo *statement* que posibilita realizar consultas sobre la BD.
- **list:** lista de post descargados de la base de datos.
- **listaHits:** lista que contiene los documentos coincidentes con un post determinado**.**
- **DB:** base de datos sobre la que trabaja la aplicación. La base de datos es ***Politiktracker***.
- **TABLE_NAME:** tabla de la base de datos de la que se van a descargar los post que han llegado más recientemente a ella. Esta tabla es ***post***.
- **HOST:** ruta del servidor al que nos conectamos mediante JDBC.
- **ACCOUNT:** el usuario con el que nos vamos a conectar a la base de datos.
- **PASSWORD:** contraseña para acceder a la base de datos.
- **DRIVER:** el driver utilizado para realizar la conexión.

Métodos principales:

- **openConnection:** método utilizado para abrir la conexión con la base de datos *Politiktracker*.
- **closeConnection:** método utilizado para cerrar la conexión con la BD.
- **descargarPostRecientes:** se trata del método que descarga los post que hayan llegado a partir de una determinada fecha (que recibe como parámetro) a la base de datos. Guarda los datos de estos post en una lista de post que ofrece como resultado.
- **obtenerPost:** método auxiliar que recibe como parámetro el Id de un post y se encarga de buscar el post correspondiente a ese Id en la tabla *post* de la base de datos. Saca como resultado los datos de ese post en un objeto tipo post.
- **obtenerNoticiaAsociada:** otro método auxiliar que dado un Id de post obtiene la noticia asociada a ese post. Lo hace acudiendo a la tabla *agrupa* de la base de datos y devuelve como resultado el código de la noticia asociada.
- **cargarNoticiaNueva:** recibe un objeto de tipo NoticiaNueva y lo inserta en la tabla *noticia* de la BD.
- **cargarPostSimilar:** recibe como parámetro un objeto de tipo PostSimilar e introduce en la tabla *agrupa* de la BD los valores necesarios.
- **eliminarNoticia:** este método recibe como parámetro un Id y elimina de la tabla *noticia* la noticia asociada a ese Id.

Nombre: *Indizador*	**Tipo:** Concreta

Superclase:	
Subclase:	
Responsabilidades:	**Colaboraciones:**
Creación del índice	
Gestión del índice añadiendo y eliminando los documentos correspondientes a las noticias de la BD.	

Atributos y variables principales:

- **w:** es el objeto de tipo *IndexWriter* que se encarga de crear y mantener la información del índice.
- **s:** es el objeto de tipo *IndexSearcher* que permite que se puedan realizar búsquedas sobre el índice creado.

Métodos principales:

- **crearIndice:** método utilizado para obtener crear el índice que servirá para buscar y relacionar noticias. La ruta del índice está fijada por defecto.
- **cerrarIndice:** es el método encargado de cerrar el índice una vez se ha terminado de trabajar con él. Antes de cerrarlo se optimiza para conseguir que las próximas búsquedas sean más eficientes y rápidas.
- **addDoc:** se encarga de guardar en el índice un objeto de tipo post que recibe como parámetro (recordar que en el índice se guarda en forma de documento).
- **add:** es el método que utiliza **addDoc** para almacenar cada campo de un documento del índice.
- **removeDoc:** método llamado por el método deleteVeryOldNews para eliminar los documentos con que lleven más de un mes en el índice.

Nombre: *ClasificaPost*	**Tipo:** Concreta
Superclase:	
Subclase:	
Responsabilidades:	**Colaboraciones:**
Realizar consultas sobre los documentos del índice	
Determinar las relaciones entre las noticias descargadas de la BD acudiendo al índice de noticias. Producir un archivo que refleje las relaciones entre los post con su grado de similitud.	

Atributos y variables principales:

- **index**: es la ruta del directorio en el que se encuentra el índice.
- **setMaxClauseCount:** parámetro que controla el tamaño del buffer necesario para hacer consultas sobre el índice.

Métodos principales:

- **buscaSimilares:** dado un objeto post y un criterio de búsqueda (por título o por texto), este método devuelve una lista de hits que se refieren a post similares al recibido.
- **asocia:** método parecido al anterior. Dado un objeto de tipo post y un criterio de búsqueda, devuelve el Id del post que más se parezca al recibido (si es que hay alguno que se parezca lo suficiente). También guarda en un archivo (*post_asociados*) las correspondencias entre los post que ha relacionado el TRACKER con el grado de similitud que tienen los post entre sí.
- **buscaIdPost:** método auxiliar utilizado por **clasificaPost** que recibe un objeto de tipo post y nos dice si ese post está dentro del índice o no.

Nombre: *Limpia*	**Tipo:** Concreta
Superclase:	
Subclase:	
Responsabilidades:	**Colaboraciones:**
Eliminar las etiquetas HTML del campo texto de las noticias que han sido descargadas de la BD.	
Borrar el índice de documentos para poder indexar los post por varios campos.	

Atributos y variables principales:

Métodos principales:

- **filtrarEtiquetasGData:** método que recibe una cadena de texto y la limpia de etiquetas HTML (recordar que el campo texto de la BD contiene estas etiquetas).
- **borrarDirectorio:** este método recibe como parámetro una cadena que determina la ubicación del directorio en el que se encuentra el índice y lo borra de manera recursiva.
- **borrarArchivo:** es el método utilizado por **borrarDirectorio** para borrar los archivos contenidos en el directorio.

Nombre: *AnalizadorCompleto*	**Tipo:** Concreta
Superclase:	
Subclase:	
Responsabilidades:	**Colaboraciones:**
Analizar los documentos que contiene el índice simplificándolos para permitir un mejor análisis.	
Almacenar la lista de *stop_words* que permiten simplificar los datos del índice.	

Atributos y variables principales:

- **SPANISH_STOP_WORDS:** es el objeto que contiene la lista de palabras que no serán tenidas en cuenta a la hora de buscar similitudes entre noticias. Estas palabras no se tienen en cuenta ya que son muy frecuentes en el lenguaje.

Métodos principales:

- **tokenStream:** método que trata el texto contenido en los documentos del índice convirtiendo las letras mayúsculas en minúsculas, eliminando las palabras que se encuentran en **SPANISH_STOP_WORDS,** y reduciendo cada palabra a su raíz semántica. Este proceso se realiza para mejorar la comparación entre noticias y no modifica en ningún momento el índice sino que trata los datos en el contenidos.
- **analizadorCompleto:** utiliza la lista de **SPANISH_STOP_WORDS** para almacenar las palabras en el objeto *stopTable* que luego será utilizado por **tokenStream**.

Nombre: *Noticia*	**Tipo:** Concreta
Superclase:	
Subclase:	
Responsabilidades:	**Colaboraciones:**
Almacenar los datos correspondientes a los campos de una noticia.	

Atributos y variables principales:

- **IdPost:** atributo destinado a guardar los datos del Id de cada noticia.
- **Numero:** atributo relacionado con el número de post relacionados con una noticia. Contendrá por defecto el valor 1 *.
- **FechaHora:** almacena la fecha y la hora a las que una noticia ha sido relacionada.

Métodos principales:

- **setIdPost:** almacena el campo Id en un objeto noticia.
- **getIdPost:** extrae el campo Id de un objeto noticia.
- **setNumero:** guarda el campo Numero en un objeto noticia.
- **getNumero:** saca el campo Numero de un objeto noticia.
- **setFechaHora: :** almacena el campo FechaHora en un objeto noticia.
- **getFechaHora:** extrae el campo FechaHora de un objeto noticia.

Nombre: *NoticiaNueva*	**Tipo:** Concreta
Superclase: *Noticia*	
Subclase:	
Responsabilidades:	**Colaboraciones:**
Copiar los datos de un objeto Post en un objeto de tipo NoticiaNueva.	

Atributos y variables principales:

Métodos principales:

- **copiarNoticiaNueva:** método que recibe un objeto de tipo post para guardarlo en un objeto tipo noticiaNueva.

Nombre: *Post*	**Tipo:** Concreta
Superclase:	
Subclase:	
Responsabilidades:	**Colaboraciones:**
Guardar los datos correspondientes a los campos de un objeto tipo post.	

Atributos y variables principales:

- **IdPost** atributo destinado a guardar los datos del Id de cada post.
- **Titulo:** destinado a almacenar los datos del título de cada post.
- **Texto:** destinado a guardar los datos de campo texto de cada post.
- **Fecha**: diseñado para guardar la fecha de cada post.

Métodos principales:

- **setIdPost:** almacena el campo Id en un objeto post.
- **getIdPost:** extrae el campo Id de un objeto post.
- **setTitulo:** se encarga de guardar el campo Título de un objeto post.
- **getTitulo:** saca el campo Título de un objeto post.
- **setTexto:** método que fija el campo Texto de un objeto de tipo post.
- **getTexto:** extrae el campo Texto de un objeto post.
- **setFecha:** guarda el campo Fecha de un objeto post.
- **getFecha:** es el método que obtiene el campo Fecha de un objeto post.

Nombre: *PostSimilar*	**Tipo:** Concreta
Superclase: *Post*	
Subclase:	
Responsabilidades:	**Colaboraciones:**
Almacenar el campo IdSimilar en un post que se asemeje a otro ya analizado.	

Atributos y variables principales:

- **IdSimilar:** es el atributo que guarda el Id de un post similar a otro analizado.

Métodos principales:

- **setIdSimilar:** método que fija el Id de un post similar.
- **getIdSimilar:** obtiene el Id de un objeto de tipo PostSimilar.

6.9. Otras consideraciones de la implementación

En el desarrollo del sistema se han utilizado algunas técnicas complementarias que deberían ser tenidas en cuenta. En primer lugar, se ha seguido un método de implementación lo más independiente posible. Esto consiste en hacer clases con métodos los más sencillos y específicos posibles.
Otro factor importante, es la utilización de ficheros de propiedades de **java** para la configuración del sistema. Inicialmente, se pueden configurar los nombres de salida, las entradas del sistema o incluso los tamaños de *buffer*. En un futuro, con este fichero se podrían configurar otros aspectos del sistema, como el tipo de salida que queremos obtener.
Se ha buscado realizar las importaciones de manera que se importen sólo las clases necesarias y no los paquetes completos. Esto permite que se ahorre una gran cantidad de tiempo de compilación. Además de las importaciones para poder trabajar con la base de datos con MySql o para poder escribir en archivos, una de las importaciones más importantes es la de **Lucene core 2.3.2** para poder trabajar con el índice.

Dinámica del sistema diseñado

Una vez se han definido todas las clases de la aplicación se puede proceder a explicar el funcionamiento dinámico del diseño realizado. Si en los anteriores diagramas y tablas se mostraban los atributos y los métodos que contenía cada clase, ahora se debe mostrar como interactúan los objetos del sistema mediante los métodos de cada uno. De esta manera los atributos de cada objeto son modificados con el funcionamiento de la aplicación.

El funcionamiento dinámico de la aplicación se aprecia muy bien con diagrama de secuencia que sigue:

Diagrama de secuencia general:

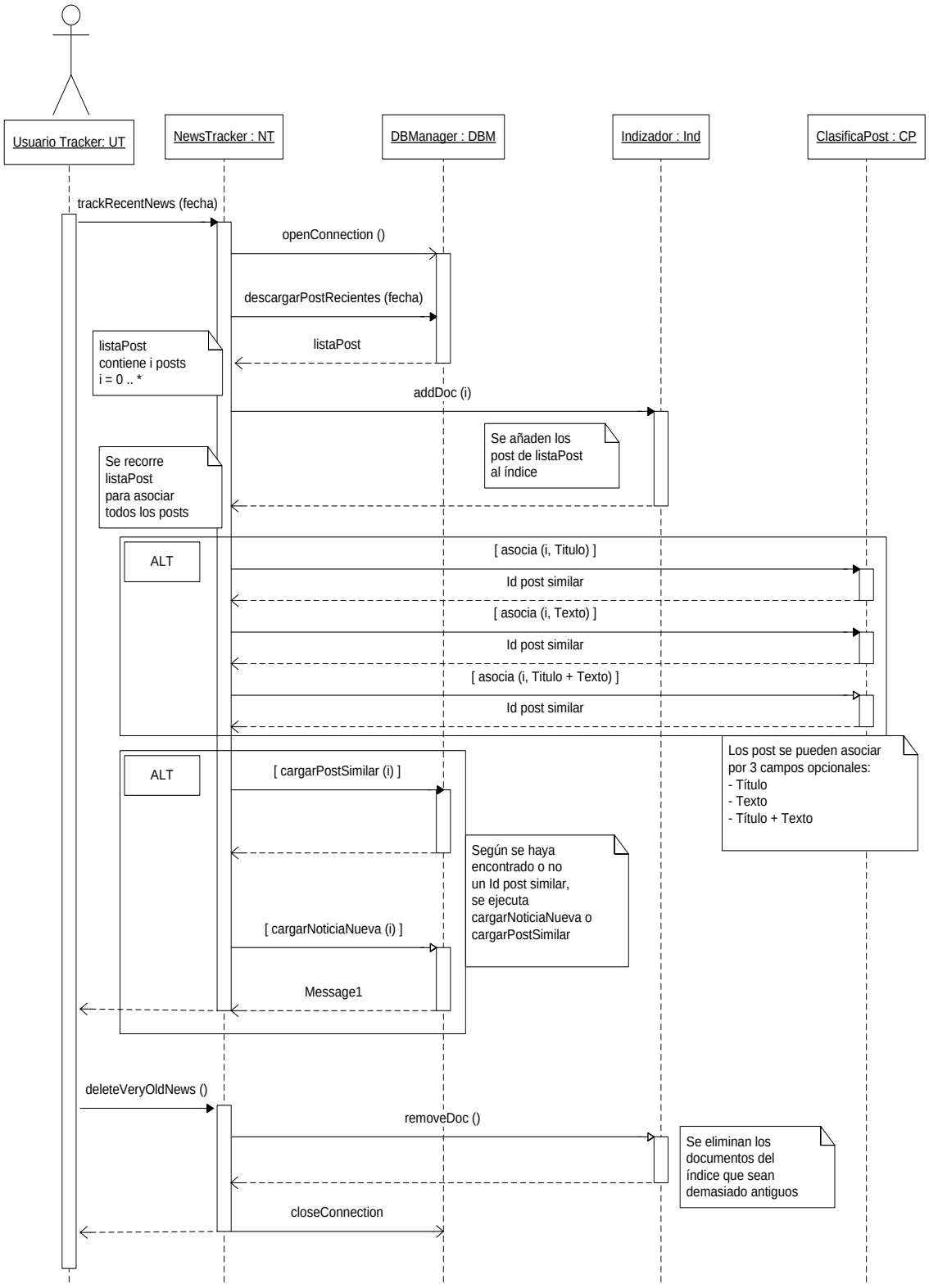


Figura 6.12: Diagrama de secuencia general

Con el diagrama de secuencia además de realizar un diseño algo más específico del funcionamiento de la aplicación, modelamos visualmente el flujo de la lógica de un sistema. Con ello conseguimos tanto documentar como validar dicha lógica.

Capítulo 7

Experimentación y resultados

La experimentación es una etapa básica en el proceso de desarrollo de software. Tras el largo proceso de estudio del entorno de la aplicación y de las tecnologías, de análisis de las necesidades que debe cubrir la aplicación, del diseño de la misma y su programación; las pruebas que se realicen sobre la aplicación determinarán si se ajusta a los requisitos establecidos, si se obtienen los resultados esperados, y lo que es más importante, se pueden extraer conclusiones de esos resultados.

En esta fase se debe tener en cuenta no sólo que la aplicación se ejecute y funcione correctamente sino que se debe buscar un nivel de exigencia alto en el análisis de los resultados. Esto se traduce en la búsqueda de un corpus de pruebas que sea interesante desde el punto de vista pedagógico y de solución de errores en la aplicación. Por tanto no se debe entender esta etapa como un mero trámite que debe pasar toda aplicación sino como una oportunidad de mejora del desarrollo y de aprendizaje del desarrollador.

Resultados negativos que denoten una falta de eficacia o efectividad en la funcionalidad de la aplicación, son más necesarios para la mejora de la aplicación que los resultados positivos que no inviten a la reflexión ni al análisis sino a la autocomplacencia. Es por esto que no hay que centrarse sólo en que los resultados de la ejecución se ajusten a los esperados, hay que buscar los puntos débiles de nuestro desarrollo y explotarlos porque esto hará más fuerte su comportamiento.

Es por todo lo anterior que resulta poco gratificante introducirse en la fase de experimentación (como en muchas otras) sin un plan previo que regule el proceso de experimentación y cambios. Es el denominado protocolo de experimentación el que debe regir la fase de pruebas para no caer en la redundancia en la obtención de resultados o la obtención de resultados poco significativos.

7.1. Protocolo de experimentación

Se han realizado pruebas modulares y pruebas de integración durante todo el proceso de construcción del sistema, estas pruebas involucran a un número creciente de módulos y terminan probando el sistema como conjunto. Al realizar estas pruebas el objetivo inicial fue determinar la variación de los valores las medidas de evaluación. Se han tratado estas pruebas como pruebas de caja blanca en las cuales se estudia el funcionamiento de la aplicación paso por paso y las relaciones de noticias que va produciendo el TRACKER. Estas pruebas diseñadas inicialmente se corresponden con los experimentos cualitativos mostrados en el apartado 7.2.1.

Las pruebas funcionales finales con el programa finalizado son pruebas de caja negra. Posteriormente se incidió en la búsqueda de los valores de las variables de la aplicación TRACKER (umbrales de similitud de título y texto) que maximizaran los valores de las medidas de evaluación. Los resultados de estas pruebas están explicados en la sección 7.2.2. de experimentos cuantitativos.

Para la evaluación cuantitativa del sistema TRACKER, dado que se calculan las medidas de evaluación de cientos de noticias, no se puede realizar manualmente. Una evaluación manual supondría un enorme gasto de tiempo y por ello se ha utilizado un módulo de evaluación externo a la aplicación que trabaja de forma automática y calcula la precisión, cobertura y f-measure con más fiabilidad y sobre todo rapidez que un ser humano.

El **módulo de evaluación** se denomina **WePS** y proviene de un proyecto de investigación de la UNED [10].

Este módulo ha sido adaptado para las necesidades de la aplicación TRACKER y se ejecuta separadamente de la aplicación. Para determinar las medidas de evaluación de un conjunto de noticias, WePS recibe un archivo (*grupos_post.xml*) que contiene los grupos de noticias creados por el sistema. Así, el módulo evaluador compara los grupos creados por el TRACKER con los grupos que deberían haberse creado si la aplicación funcionara perfectamente (como si la agrupación la realizara un humano) que denominaremos *grupos_referencia.xml*. Hay que recordar que para determinar la precisión y la cobertura de un grupo de noticias hace falta tener ese archivo de referencia para el grupo de noticias que se debe hacer de forma manual por un ser humano. De esta manera el evaluador obtiene las medidas de evaluación requeridas.

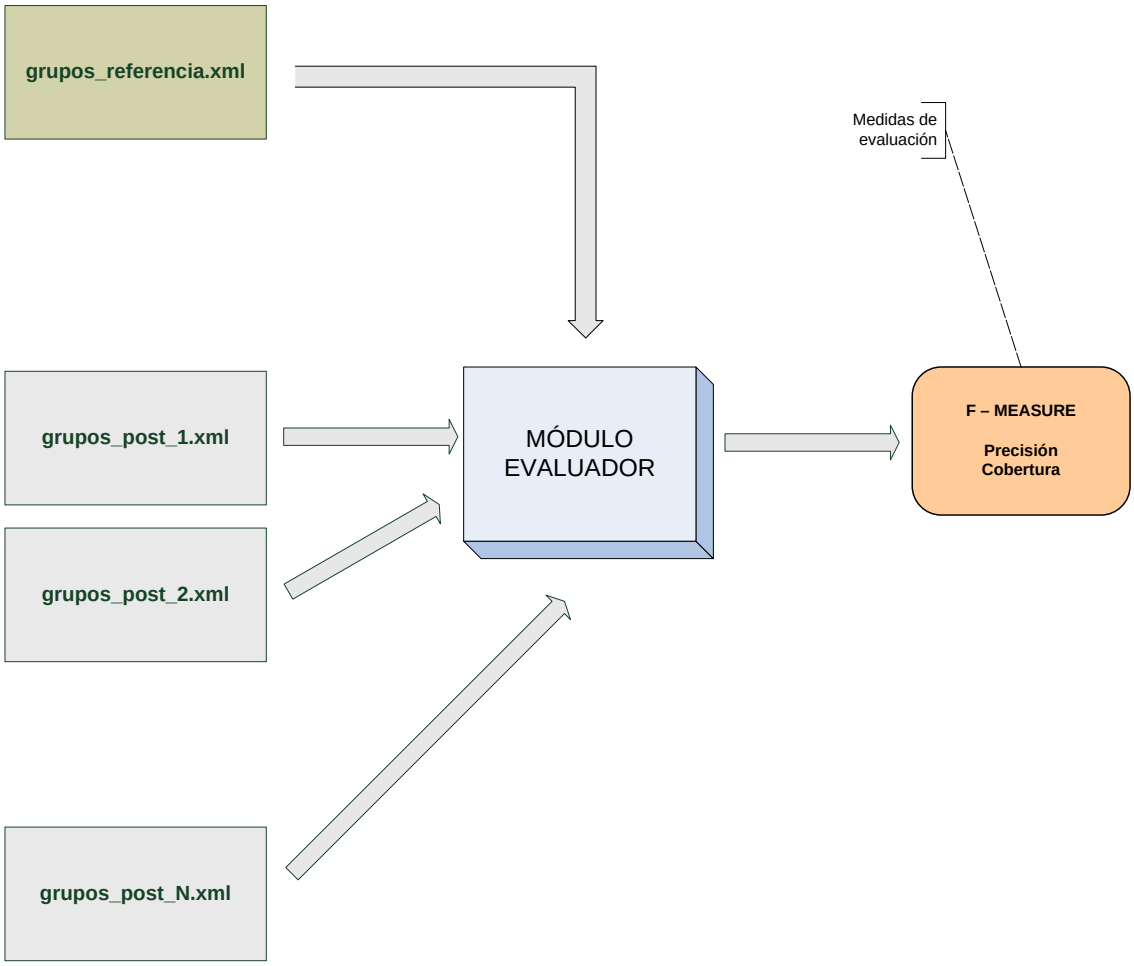


Figura 7.1: Funcionamiento del módulo de evaluación

Donde los diferentes grupos de post (grupos_post_1, grupos_post_2, … , grupos_post_N) son los grupos de post creados por el TRACKER dependiendo de un parámetro que se puede variar en la aplicación. Esta medida es el **Umbral de Similitud** (US) y se fija antes de la ejecución del tracker. Este umbral es el explicado en el apartado 7.2 de la presente memoria, y fija el porcentaje de similitud a partir del cual la aplicación va a relacionar dos noticias. Para nuestro sistema se puede fijar un umbral de similitud para el campo título – US (título) – y para el campo texto – US (texto) – ya que las noticias se relacionan por ambos campos.

Si se fija US (título) = 0,05 significa que en *grupos_post.xml* sólo se guardarán grupos de noticias que se relacionen entre sí con una similitud de su título mayor o igual del 5%. De la misma manera para el US (texto).

Para la evaluación de la aplicación hay que prestar especial atención a la definición de tópico puesto que es un tanto ambigua, y a veces es difícil distinguir cuándo un conjunto de sucesos forman un único tópico o una secuencia de tópicos. Por ejemplo, en el caso de un accidente con el conductor a la fuga se producen varios sucesos o sub-sucesos: el accidente, la detención del conductor y el juicio. Otro ejemplo podrían ser los episodios o distintas partes de una guerra: comienzo del conflicto, guerra, acuerdos de paz, etc. ¿Forman todos estos sucesos parte del mismo tema o son varios sucesos en cascada, es decir una secuencia de tópicos? En nuestro sistema vamos a suponer como norma general que dos sucesos forman parte del mismo tópico o tema si uno es la reacción del otro (que sus acciones se desarrollen próximas en el tiempo).

Para la evaluación de los parámetros propuestos para aplicación de relación de noticias, se han clasificado manualmente los documentos que están almacenados en la BD *Politiktracker*. Pero no se han tomado todos los documentos de la BD ya que resultaría muy difícil clasificar tantos documentos manualmente y sobre todo conllevaría un larguísimo espacio de tiempo. En nuestro caso se utilizarán las medidas que se proponen en TDT y que son universales en cuanto a que son las utilizadas como referencia en todos los trabajos de TDT.

Estas medidas o variables se muestran a continuación:

$$\text{Precisión} = \frac{documentos\_relevantes\_recuperados}{documentos\_recuperados}$$

$$Cobertura = \frac{documentos\_relevantes\_recuperados}{documentos\_relevantes}$$

Y sobre todo, la medida combinada de las dos anteriores. Esta medida es básica pues permite maximizar su valor teniendo en cuenta la cobertura y la precisión. La F-Measure definida en el módulo evaluador viene dada por la siguiente ecuación:

$$F-Measure = \frac{1}{\frac{\alpha}{precisión} + \frac{(1-\alpha)}{cobertura}}$$

Para evaluar nuestro sistema cualitativamente (en los experimentos cualitativos) se emplearán principalmente las dos primeras medidas (cobertura y precisión) que permiten diagnosticar el funcionamiento de la aplicación de manera bastante fiable y obteniendo una impresión mucho más intuitiva del trabajo del TRACKER.

Sin embargo, para valorar el sistema cuantitativamente (dentro de los experimentos cuantitavos) se empleará la medida F-Measure que otorga mucha más exactitud a la valoración y una combinación equitativa de precisión y cobertura.

Hay que entender que para la evaluación del TRACKER primero hay que realizar una clasificación manual de las noticias y por ello se indican los criterios que se han seguido en el proceso de comparación entre la clasificación manual de noticias y la clasificación que realiza la aplicación. Estos criterios son básicos pues un cambio en los mismos supone un cambio en las medidas de precisión y cobertura de los grupos de noticias.

- Una noticia se considera bien relacionada con otra si ambas noticias pertenecen al mismo grupo de referencia determinado manualmente.
- Un grupo de noticias generado por el TRACKER será encuadrado dentro del tema al que pertenezcan más noticias del grupo.
- Las noticias que no han sido relacionadas con ninguna otra por la aplicación, tendrán la consideración de temas individuales. O lo que es lo mismo, serán considerados grupos de una sola noticia.

7.2. Experimentos

A continuación se muestran los experimentos realizados para la evaluación del sistema de relación de noticias políticas. Todos los experimentos tienen en común el origen del corpus de datos de trabajo que es la base de datos *Politiktracker*. Estas noticias que componen el corpus de trabajo, se almacenaron en la BD mediante la aplicación *Crawler* (traslada la información de RSS de páginas de periódicos y blogs a la BD) que es otro de los módulos del proyecto general MEMETRACKER. Sin este *Crawler* sería imposible el trabajo de la aplicación TRACKER ya que no habría noticias que relacionar. Lo que varía en cada experimento es el número y fecha de los post escogidos para ser tratados.

Además del rango de fechas a partir del cual se realiza la ejecución también se ha modificado para cada experimento el umbral de similitud (US) a partir del cual se relacionan las noticias automáticamente. Dado que se realiza la relación de noticias por los campos título y texto, el US puede ser diferente en los experimentos para ambos campos.

La sección de experimentos se divide en dos apartados:

- En primer lugar se presenta un análisis experimental de tipo cualitativo en el que se muestran las diferentes asociaciones que se producen entre las noticias según el umbral de similitud. En este apartado son de especial relevancia los gráficos que muestran como aumenta o disminuye el número de noticias dentro de los grupos. En este apartado no se busca conseguir una gran eficiencia del TRACKER en cuanto a precisión y cobertura sino que se trata de explicar la importancia de cada elemento y variable dentro del proceso de seguimiento y detección de noticias.
- El segundo apartado de la sección enfoca los experimentos desde el punto de vista cuantitativo. En este caso se trata de obtener el mayor número de datos posible para poder extraer conclusiones sobre los valores más adecuados de los parámetros del tracker. En este apartado no se profundiza en como se relacionan las noticias, asunto tratado en el apartado de experimentos cualitativos, sino que se hace hincapié en la tendencia de la F-Measure, la cobertura y la precisión cuando se varían los valores del umbral de similitud. En estos experimentos lo importante es sacar conclusiones sobre el valor adecuado de los US.

7.2.1. Experimentos cualitativos

A continuación se procede a explicar los tres experimentos en los que se han analizado las relaciones entre noticias.

7.2.1.1. Experimento 1.1.

Este primer experimento supone la ejecución del TRACKER a partir de las 16 horas del 13 de octubre de 2008 sobre el conjunto de post descargados hasta ese momento en la base de datos.

En total se trabaja sobre un corpus de 118 noticias de las cuales se determinan manualmente 78 temas de los cuales 11 están compuestos por más de una noticia y el resto son temas unitarios con un solo documento representativo.

Para la ejecución de la aplicación se ha fijado un umbral de similitud del 5 % tanto para el campo título como para el campo texto. Es decir US = 0.05.

7.2.1.1.1. Generación de un corpus de evaluación.

Se han identificado 11 temas comunes o clusters que contienen más de una noticia.

Estos clusters son los siguientes:

CLUSTER	TEMA	Nº de noticias asociadas al tema
1	Elecciones EE.UU.	5
2	Armstrong correrá el Giro	2
3	Detención de Zigor Goieskoetxea	3
4	Desfile del día de la Hispanidad + Rajada de Rajoy	10
5	Manifestaciones día de la Hispanidad	5
6	Lluvias y temporal en Murcia	3
7	Conflicto PP-UPN	4
8	Crisis financiera	14
9	Plan Eurozona	3
10	Reconocimiento a Companys	3
11	Premios Nobel	2

El campo *Nº de noticias asociadas al tema* será clave para poder medir la cobertura de un determinado tema que ha realizado el programa.

Se debe recordar que los grupos de noticias no son los que ha realizado la aplicación sino que son los determinados por el ser humano. En este caso los temas han sido determinados por el autor del proyecto y es posible que otra persona hiciera una agrupación algo diferente.

La identificación de temas y por tanto la relación de noticias es una tarea con un gran componente subjetivo, sin embargo la mayoría de noticias tratadas aún presentando cierta ambigüedad serían clasificadas de la misma manera por cualquier persona con un mínimo conocimiento de la actualidad.

En nuestro caso, los temas son bastante diferentes en cuanto a alcance y duración del acontecimiento, pero sólo se han encontrado dudas en cuanto a los temas de la "Crisis financiera" (8) y del "Día de la Hispanidad" (4).

En el caso de la crisis se trata de un asunto de amplio calado a todos los niveles y se ha decidido agrupar en este tema noticias que refirieran las medidas, consecuencias, reacciones, etc. ante la crisis económica. Se trata de un tema que tiene un alcance mundial y muy prolongado en el tiempo, es por esto que al detectar algunas noticias sobre el "Plan Eurozona" (9) se haya decidido agrupar éstas en un nuevo tema ya es un subtema muy concreto y aunque derivado de la crisis está suficientemente diferenciado.

La otra diferenciación complicada en el corpus de noticias es la del día de la Hispanidad, tema que podría haber sido considerado un único tema junto con el tema "Manifestaciones día de la Hispanidad" (5). En este caso se ha optado por la separación siguiendo un criterio espacial ya que el desfile y las manifestaciones fueron en lugares muy distantes y son acontecimientos muy concretos.

7.2.1.1.2. Resultados

El programa realiza las agrupaciones en el fichero de `post_asociados.txt` dando lugar a 62 relaciones de noticias como se especifica en la siguiente tabla.

La agrupación es el campo que indica que post a agrupado la aplicación como pertenecientes a la misma temática. El campo cluster es el que refleja a que cluster pertenece el mayor número de noticias de la agrupación. Los campos cobertura y precisión son las medidas correspondientes para la agrupación hecha por el TRACKER respecto a la agrupación perfecta (Si hay una cobertura del 100% y también una precisión del 100% significa que ese grupo de posts se ha agrupado perfectamente).

En el campo agrupación se pueden apreciar algunos identificadores de post en color rojo. Esto significa que dichos post no están bien relacionados por la aplicación en ese grupo de noticias, lo que restará precisión al grupo.

AGRUPACIÓN	CLUSTER	COBERTURA (%)	PRECISIÓN (%)
40950 - 40951 - 41014 - 41177 - 41216	1	100	100
40968 - 41097 - 41281	8	21,43	100
41012 - 41133 - 41434	9	100	100
41070 - 41189 - 41425	3	100	100
41085 - 41086 - 41121- 41198 - 41201 - 41219 - 41220 - 41221	4	80	100
41089 - 41192 - 41200	5	60	100
41092 - 41190 - 41196	10	100	100
41096 - 41123 - 41421	8	14,28	66,66
41125 - 41173 - 41211	11	100	66,66
41174 - 41367 - 41431	7	75	100
41180 - 41191 - 41193 - 41199 - 41435 - 41436	4	20	33,33
41202 - 41422 - 41440	6	66,66	66,66
41013 - 41428	2	100	100
41071 - 41429		0	0
41073 - 41195		0	0
41091 - 41183	8	15,38	100
41128 - 41129		0	0
41178 - 41214		0	0
41215 - 41439		0	0
41225 - 41423	8	14,28	100

Para un total de 62 de noticias relacionadas en grupos de más de una noticia se obtienen los siguientes datos ponderados.

Cobertura (grupos) = 56,08 %

Precisión (grupos) = 72,57 %

Pero no hay que olvidar que además de las relaciones mostradas en `post_asociados.txt` también hay otras 56 noticias que no han sido relacionadas con ningún tema y de las cuales también hay que estudiar las medidas de precisión y

cobertura. Obviamente la precisión para todas estas noticias va a ser del 100% pero la cobertura puede ser menos si ese post debía haber sido incluido en algún cluster por la aplicación.

En la siguiente tabla se muestran las noticias que deberían haber sido incluidas en algún cluster y por tanto tienen una cobertura menor del 100%.

POST	CLUSTER	COBERTURA (%)
41093	8	7,14
41132	5	20
41176	8	7,14
41197	5	20
41218	8	7,14
41349	8	7,14
41427	6	33,33
41432	7	25
41433	8	7,14

El resto de post no relacionados por el *tracker* conforman grupos unitarios de noticias, es decir, el programa ha hecho bien al dejarlos sin relacionar con ningún otro post. Todos estos grupos unitarios tienen cobertura y precisión del 100%.

Los resultados para el total de grupos unitarios detectados por el tracker son:

Cobertura (unitarios) = 86,32 %

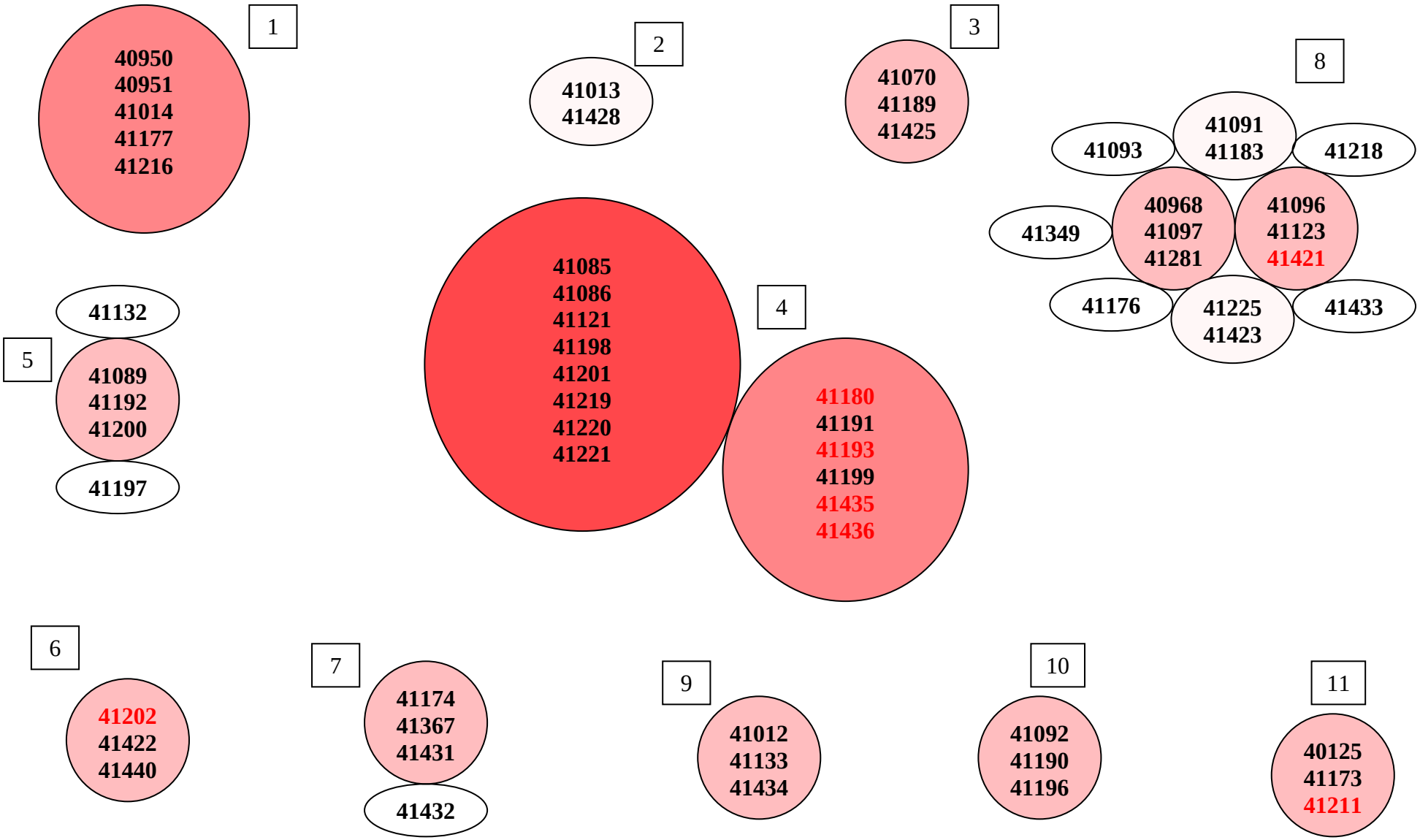
Precisión (unitarios) = 100 %

Incluyendo también los post no relacionados por el *tracker* en la medida de la cobertura y la precisión, se obtiene que para el total de 118 posts tratados por la aplicación los valores de cobertura y precisión son:

Cobertura (final)	**=**	**70,43 %**
Precisión (final)	**=**	**85,59 %**

Cabe reseñar que para el total de 118 posts, el programa *trcker* ha colocado erróneamente 17 posts, bien agregándolos a algún cluster (7 posts) o relacionándolos entre sí (10 posts).

CLUSTERS TRAS LA EJECUCIÓN
1.1.



A continuación se muestran los datos de la cobertura y la precisión para cada uno de los 11 clusters de noticias.

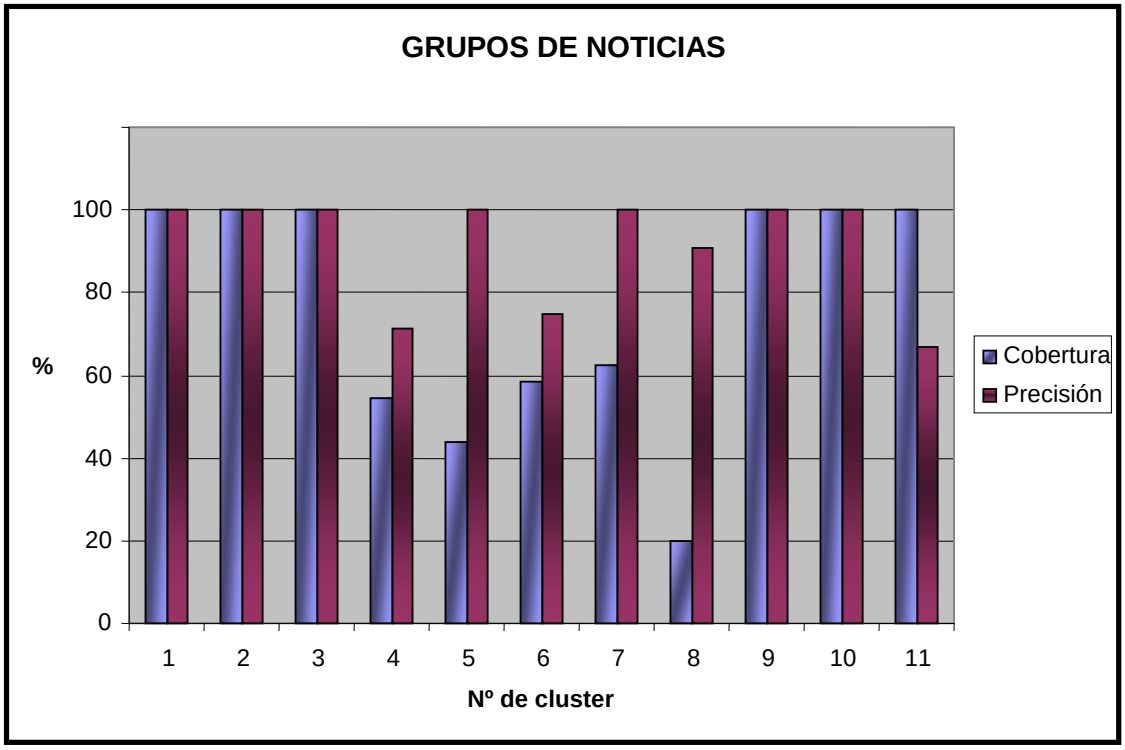


Figura 7.2: Precisión y cobertura de cada cluster del experimento 1.1.

7.2.1.1.3. Interpretación

De los datos que se han mostrado en la sección de resultados de este experimento se pueden sacar varias conclusiones. En primer lugar hay que realizar una diferenciación entre los datos obtenidos para los grupos de varias noticias y los datos de grupos temáticos unitarios (de una sola noticia).

Grupos de varias noticias

Para los grupos de más de una noticia creados por el *tracker* se ha determinado que la cobertura y la precisión eran respectivamente del 64,10 % y 74,57 % respectivamente.

La explicación del valor algo reducido de la cobertura es que el *tracker* ha restringido demasiado las relaciones entre la noticias. Esta restricción se regula con el umbral de similitud que para este experimento es US = 0.05 tanto para el campo título como para el campo texto. Disminuyendo el umbral de similitud (el de alguno de los dos campos) se conseguiría que el tracker relacionase más noticias entre sí aumentando la cobertura.

Sin embargo, esa disminución del US con el consiguiente aumento del número de relaciones, traería consigo una disminución de la precisión ya que se produciría más relaciones erróneas que con el US actual.

Por otro lado, la explicación del valor de los fallos que se han producido en algunas relaciones y que han reducido la precisión para el experimento, está en algunas palabras en concreto que son confusas y resultan ambiguas para la aplicación.

> *Palabras que confunden las noticias en el texto principalmente: Zapatero, Rajoy, diputados, premios, Marcha (diferencia entre manifestación y desfile), Nacional, Parlamento, reforma, acto, Dios, española…*

Como se observa, las palabras que confunden al tracker son principalmente nombres. Hay que recordar que el funcionamiento del programa se basa en un algoritmo que filtra el contenido de las noticias. Uno de esos filtros es el filtro de *stop words* que se encarga ignorar las palabras que sean demasiado comunes en el texto a analizar. Podría resultar tentador incluir algunas de las palabras que han confundido al tracker en la lista de stop words para que no fueran tenidas en cuenta…

Pero esto es imposible ya que los nombres son básicos para determinar el sentido de un texto (tanto para una máquina como para un ser humano) y por esto, si se incluyese la palabra “Zapatero” en la lista, se vería resentido el funcionamiento del programa para el conjunto de noticias que se refiriesen a un acto de Zapatero. Probablemente solucionaría un caso en concreto de relación mal determinada, pero a costa de perjudicar otras muchas relaciones y el funcionamiento general.

Por tanto, después de un análisis profundo de cada relación se debe concluir que los fallos cometidos por el tracker no son subsanables a no ser que se incluyan en la aplicación otras funcionalidades como la detección de términos y expresiones temporales. Ningún cambio en los parámetros regulables del tracker mejoraría ostensiblemente la precisión obtenida.

Grupos unitarios de noticias

Para el conjunto de los grupos unitarios mejoraron enormemente la cobertura y la precisión (recordar que la precisión para un solo post siempre va a ser el 100 %.Esto es debido a que toda noticia pertenece a algún tema y por tanto una noticia no relacionada se considerará como perteneciente al tema oportuno).

La cobertura experimenta un gran aumento ya que hay muchos post que representan en sí mismo una noticia y no hay en el resto del corpus ningún post que trate el mismo tema. Esto se debe a que la base de datos alberga noticias de temática muy variada y en los 118 post estudiados en este experimento se ve reflejada esta situación.

Si el tracker hubiera trabajado sobre un conjunto de post que se hubieran descargado en la BD tras un gran acontecimiento que hubiera copado todos los medios de comunicación en Internet, esto se hubiera visto reflejado en la cobertura. Con un gran acontecimiento se reducen las noticias de otros temas, y en casi todas las páginas de periódicos, blogs y otros sitios de la red se escriben noticias sobre ese tema. En este caso el tracker vería resentido su rendimiento ya que debería realizar muchas más relaciones reduciéndose ciertamente la cobertura y muy probablemente la precisión.

Los 48 posts individuales que se indican a la derecha se corresponden con noticias individuales. Estos posts son los individuales que tienen cobertura y precisión del 100 %, no así los individuales agregados a algún cluster como se ha explicado anteriormente.

Sin embargo, en esta lista de debería haber 10 posts más ya que el tracker ha realizado 5 relaciones binarias de manera incorrecta dentro de las cuales ningún post se corresponde con algún grupo temático.

En el cuadro inferior se detallan estas 5 relaciones. Por supuesto éstas tendrán una precisión y cobertura del 0 % ya que no deberían haber sido relacionadas.

48 NOTICIAS INDIVIDUALES

40997	41124	41208
40998	41126	41209
40999	41127	41210
41000	41130	41212
41019	41131	41213
41021	41134	41217
41022	41175	41222
41072	41179	41223
41074	41181	41224
41087	41182	41384
41088	41184	41412
41090	41185	41424
41093	41186	41426
41094	41187	41430
41095	41188	41437
41122	41194	41438

5 RELACIONES BINARIAS ERRÓNEAS

41071 – 41429
41073 – 41195
41128 – 41129
41178 – 41214
41215 – 41439

Análisis de los clusters de noticias

En a gráfica del punto 7.2.1.2 llamada “Grupos de noticias” se muestra gráficamente cómo han sido las relaciones para los diferentes clusters. Con esta gráfica se observa que hay 5 grupos temáticos de noticias para los que el tracker ha funcionado a la perfección obteniéndose la cobertura y precisión máximas. Hay otros 5 grupos de noticias para los que la cobertura es menor que la precisión. Finalmente hay un tema para el que la precisión resulta menor que la cobertura.

En el conjunto de los **grupos de precisión y cobertura perfectas** sólo hay que destacar que se trata de **temas muy concretos** como “Armstrong correrá el Giro”, “Detención de Zigor Goieskoetxea” o “Reconocimiento a Copanys”, o bien se trata de temas referentes al extranjero como son “Elecciones de EE.UU.” o “Plan Eurozona”. La característica común a estos grupos de noticias es que en todas las noticias de los grupos se utilizan nombres propios muy específicos. Esto favorece el funcionamiento del tracker puesto que las noticias no dan lugar a muchas ambigüedades.

En el conjunto de los grupos que tienen menor cobertura que precisión hay que destacar en especial dos temas ya que son representativos de la situación en que *cobertura* < *precisión*, “Desfile día de la Hispanidad” y “Crisis financiera”. Se trata de temás bastante generales y que contienen más noticias que otros clusters. Esto produce que ante un umbral de similitud no demasiado bajo, se formen subgrupos de noticias dentro del propio grupo. Los subgrupos no tienen porque afectar a la medida de la precisión si las relaciones están bien construidas, pero lo que seguro provocarán es una disminución de la cobertura.

El último grupo referente al tema de “Premios Nobel” no resulta demasiado representativo ya que consta sólo de 2 noticias. Esto hace que un fallo en alguna relación produzca una disminución muy grande de la precisión en relación a la cobertura. En este caso se ha relacionado una noticia referente a otro tipo de premios con este grupo.

7.2.1.2. Experimento 1.2.

7.2.1.2.1. Generación de un corpus de evaluación

En este experimento se trabaja con el mismo corpus de noticias que en el experimento 1.1, es decir, se tratan las noticias descargadas de la BD Politiktracker a partir del 13 de octubre de 2008 a las 16 horas. La única diferencia con el primer experimento cualitativo está en los umbrales de similitud, ya que son ambos inferiores para este.

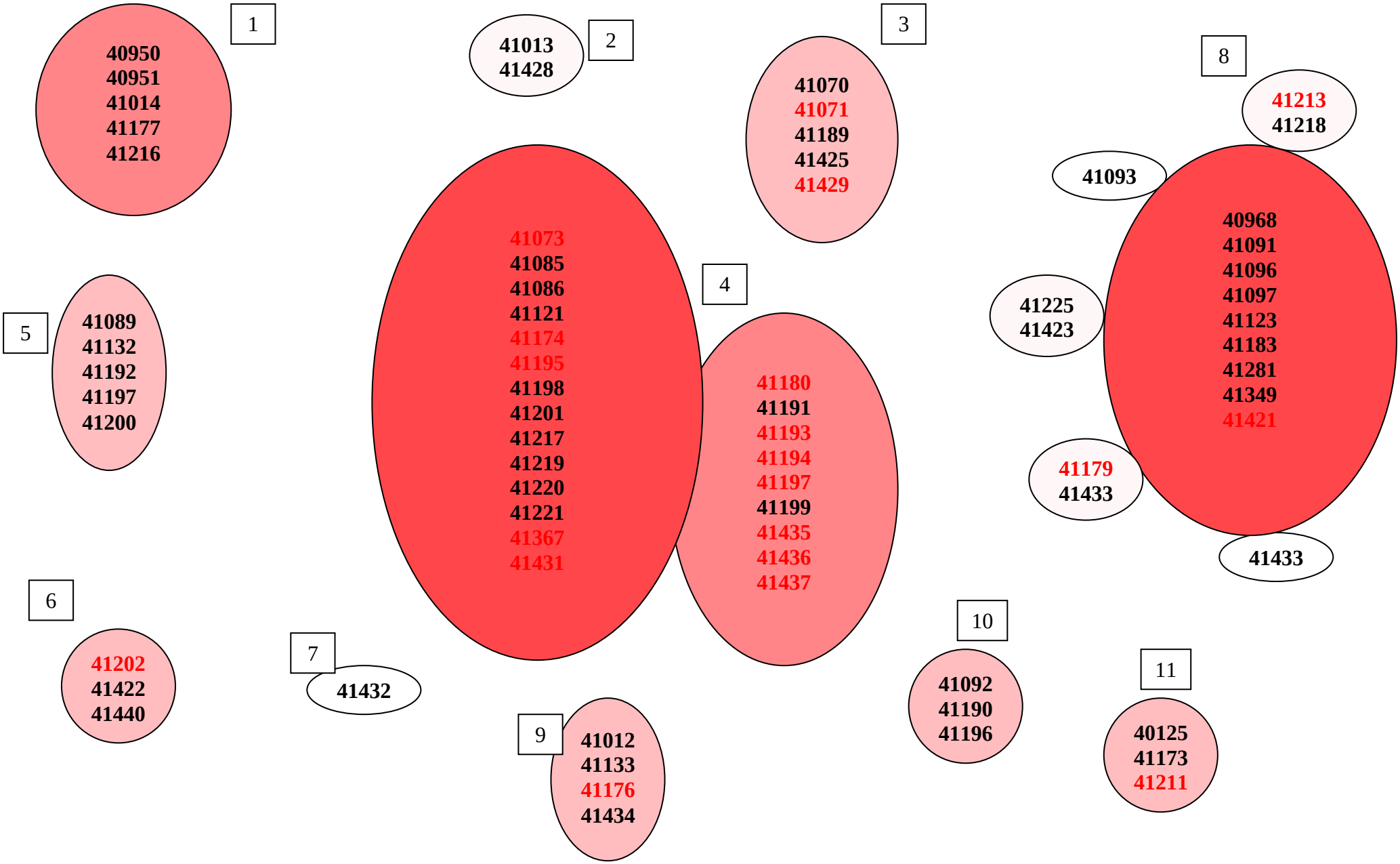
US (título) = 0,03
US (texto) = 0,03

7.2.1.2.2. Resultados

El programa realiza las agrupaciones en el fichero de `post_asociados.txt` dando lugar a 78 relaciones de noticias como se especifica en los gráficos.

Al igual que en el experimento 1.1 se pueden apreciar algunos identificadores de post en color rojo. Esto significa que dichos post no están bien relacionados por la aplicación en ese grupo de noticias, lo que restará precisión al grupo.

CLUSTERS TRAS LA EJECUCIÓN
1.2.



39 NOTICIAS INDIVIDUALES

40997	41124	41208
40998	41126	41209
40999	41127	
41000	41130	
41019	41131	
41021	41134	
41022	41175	41222
41072		41223
41074	41181	41224
41087	41182	41384
41088	41184	41412
41090	41185	41424
41093	41186	41426
	41187	41430
	41188	
41122		41438

5 RELACIONES BINARIAS ERRÓNEAS

41094 – 41095 -
41128 – 41129
41178 – 41214
41210 – 41212 -
41215 – 41439

- Entran 2 nuevas relaciones binarias erróneas y salen otras 2. Se queda como estaba en 10 relaciones binarias erróneas.
- 9 de las noticias individuales se unen a algún grupo (erróneamente)

Como se observa, respecto al experimento original se han reducido el número de grupos formados para aumentar en tamaño. Al Haber disminuido el US, se han producido también más relaciones erróneas.

AGRUPACIÓN	CLUSTER	COBERTURA (%)	PRECISIÓN (%)
40950 - 40951 - 41014 - 41177 - 41216	1	100	100
40968 - 41091 - 41096 - 41097 - 41123 - 41183 - 41281 - 41349 - 41421	8	57,14	88,88
41012 - 41133 - 41176 - 41434	9	100	75
41070 - 41071 - 41189 - 41425 - 41429	3	100	60
41073 - 41085 - 41086 - 41121 - 41174 - 41195 - 41198 - 41201 - 41217 - 41219 - 41220 - 41221 - 41367 - 41431 -	4	90	64,28
41089 - 41132 - 41192 - 41197 - 41200	5	100	100
41092 - 41190 - 41196	10	100	100
41125 - 41173 - 41211	11	100	66,66
41180 - 41191 - 41193 - 41194 - 41197 - 41199 - 41435 - 41436 - 41437	4	20	22,22
41202 - 41422 - 41440	6	66,66	66,66
41013 - 41428	2	100	100
41213 - 41218	8	7,14	50
41225 - 41423	8	14,28	100
41179 - 41433	8	7,14	50
41094 - 41095		0	0
41128 - 41129		0	0
41178 – 41214		0	0
41210 – 41212		0	0
41215 – 41439		0	0

Para un total de 78 de noticias relacionadas en grupos de más de una noticia se obtienen los siguientes datos ponderados.

Cobertura (grupos) = 62,97 %

Precisión (grupos) = 61,54 %

POST	CLUSTER	COBERTURA (%)
41093	8	7,14
41432	7	25
41433	8	7,14

Los resultados para el total de grupos unitarios detectados por el tracker son:

Cobertura (unitarios) = 93,48 %

Precisión (unitarios) = 100 %

Incluyendo también los post no relacionados por el tracker en la medida de la cobertura y la precisión, se obtiene que para el total de 118 posts tratados por la aplicación los valores de cobertura y precisión son:

Cobertura (final) = 73,31 %

Precisión (final) = 74,58 %

Cabe reseñar que para el total de 118 posts, el programa *trcker* ha colocado erróneamente 30 posts, bien agregándolos a algún cluster (20 posts) o relacionándolos entre sí (10 posts).

7.2.1.2.3. Interpretación

La disminución de los umbrales de similitud provoca que aumente el número de relaciones, pero esto se produce a costa de reducir la precisión en las relaciones.
Lógicamente aumenta la cobertura, como cabe esperar tras la disminución de la precisión. Hay que recordar que son valores totalmente ligados y que un aumento de la precisión conlleva una disminución de la cobertura y viceversa. También aumenta la tasa de fallo del tracker, esto es, aumenta el número de relaciones erróneas entre noticias.

7.2.1.3. Experimento 1.3.

7.2.1.3.1. Generación de un corpus de evaluación

Para este experimento se continúa trabajando con el mismo corpus de noticias descargado el 13-10-2008 a las 16:00 de la misma manera que en los experimentos 1.1 y 1.2. La única diferencia está en los umbrales de similitud, que en este caso son los mayores para los experimentos cualitativos.

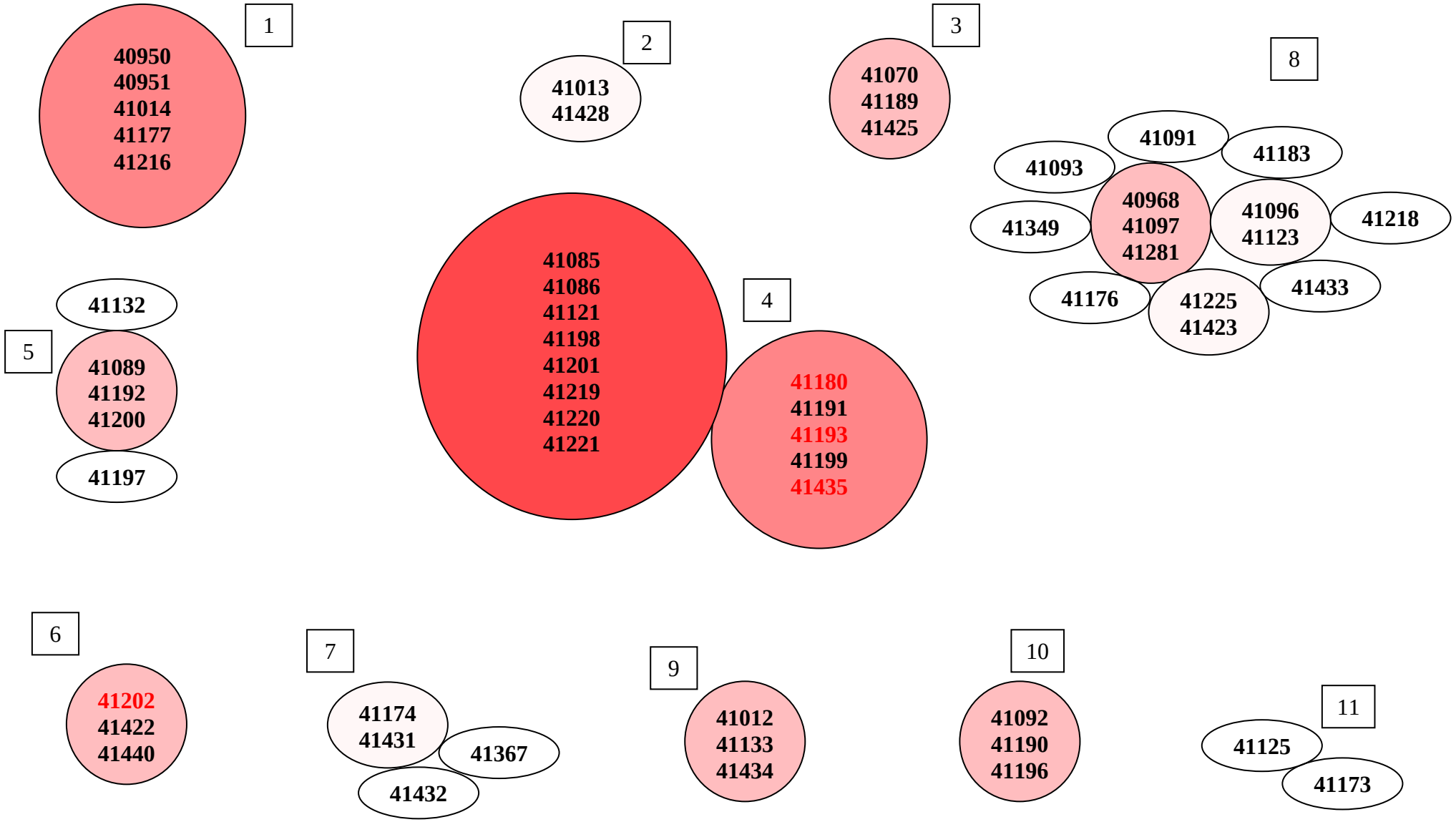
US (título) = 0,07
US (texto) = 0,07

7.2.1.3.2. Resultados

En este experimento el programa es más exigente al realizar la relación entre noticias ya que los US han aumentado considerablemente. Se producen 50 relaciones de noticias como se especifica en los gráficos.

Los identificadores de post en color rojo indican que dichos post no están bien relacionados por la aplicación en ese grupo de noticias, lo que restará precisión al grupo.

CLUSTERS TRAS LA EJECUCIÓN
1.3



51 NOTICIAS INDIVIDUALES

40997	41124	41208
40998	41126	41209
40999	41127	41210
41000	41130	41212
41019	41131	41213
41021	41134	41217
41022	41175	41222
41072	41179	41223
41074	41181	41224
41087	41182	41384
41088	41184	41412
41090	41185	41424
41093	41186	41426
41094	41187	41430
41095	41188	41437
41122	41194	41438
41211		
41421		
41436		

3 RELACIONES BINARIAS ERRÓNEAS

41073 – 41195
41128 – 41129
41178 – 41214

- Se crean 2 relaciones binarias erróneas.
- Se almacenan correctamente 3 nuevas noticias individuales respecto al experimento 1.1.

AGRUPACIÓN	CLUSTER	COBERTURA (%)	PRECISIÓN (%)
40950 - 40951 - 41014 - 41177 - 41216	1	100	100
40968 - 41097 - 41281	8	21,43	100
41012 - 41133 - 41434	9	100	100
41070 - 41189 - 41425	3	100	100
41085 - 41086 – 41121 - 41198 - 41201 - 41219 - 41220 - 41221	4	80	100
41089 - 41192 - 41200	5	60	100
41092 - 41190 - 41196	10	100	100
41096 - 41123	8	14,28	100
41174 - 41431	7	50	100
41180 - 41191 - 41193 - 41199 - 41435	4	20	40
41202 - 41422 - 41440	6	66,66	66,66
41013 - 41428	2	100	100
41225 - 41423	8	14,28	100
41073 - 41195		0	0
41128 - 41129		0	0
41178 - 41214		0	0

Para un total de 50 de noticias relacionadas en grupos de más de una noticia se obtienen los siguientes datos ponderados.

Cobertura (grupos) = 58,83 %

Precisión (grupos) = 80,00 %

POST	CLUSTER	COBERTURA (%)
41091	8	7,14
41093	8	7,14

41125	11	50
41132	5	20
41173	11	50
41176	8	7,14
41183	8	7,14
41197	5	20
41218	8	7,14
41349	8	7,14
41432	7	25
41433	8	7,14

Los resultados para el total de grupos unitarios detectados (los 12 anteriores más los que forman noticias individuales) por el tracker son:

Cobertura (unitarios) = 49,28 %

Precisión (unitarios) = 100 %

Incluyendo también los post no relacionados por el tracker en la medida de la cobertura y la precisión, se obtiene que para el total de 118 posts tratados por la aplicación los valores de cobertura y precisión son:

Cobertura (final) = 53,33 %

Precisión (final) = 91,52 %

Cabe reseñar que para el total de 118 posts, el programa tracker ha colocado erróneamente 10 posts, bien agregándolos a algún cluster (4 posts) o relacionándolos entre sí (6 posts).

7.2.1.3.3. Interpretación

Para este experimento la cobertura resulta manifiestamente inferior que en los otros dos experimentos. Esto es consecuencia lógica de la gran dificultad a la hora de relacionar las noticias. Al aumentar los US de título y texto se producen muchas menos relaciones y por tanto se cubren mucho peor todos los grupos de noticias. A cambio se produce un aumento de la precisión ya que se arriesga menos al hacer las relaciones y por tanto la mayoría son correctas.

7.2.1.4. Comparativa de los experimentos cualitativos

Con la siguiente gráfica únicamente se trata de identificar la tendencia de la cobertura y la precisión según el umbral de similitud elegido.

	Exp. 1.1. (US titulo = 0,05) (US texto = 0,05)	**Exp. 1.2. (US titulo = 0,03) (US texto = 0,03)**	**Exp. 1.3. (US titulo = 0,07) (US texto = 0,07)**
Cobertura (%)	70,43	73,31	53,33
Precisión (%)	85,59	74,58	91,52

Como se observa, la tendencia es que al aumentar el US aumente también la precisión pero disminuya la cobertura. Lo idóneo es encontrar un equilibrio entre ambas para lograr cubrir la máxima cantidad de temas con la mayor precisión posible.

En cualquier caso, para nuestra relación de noticias es preferible conseguir una alta cobertura antes que una gran precisión. Esto es debido a que el usuario está interesado en encontrar todas las noticias relacionadas con un tema y si no las obtiene todas por la aplicación le resultará casi imposible encontrarlas en los diferentes medios. Sin embargo, las noticias sobrantes de cada grupo de noticias son fáciles de filtrar por el usuario que con sólo un vistazo puede diferenciar si una noticia está relacionada o no con la temática del grupo.

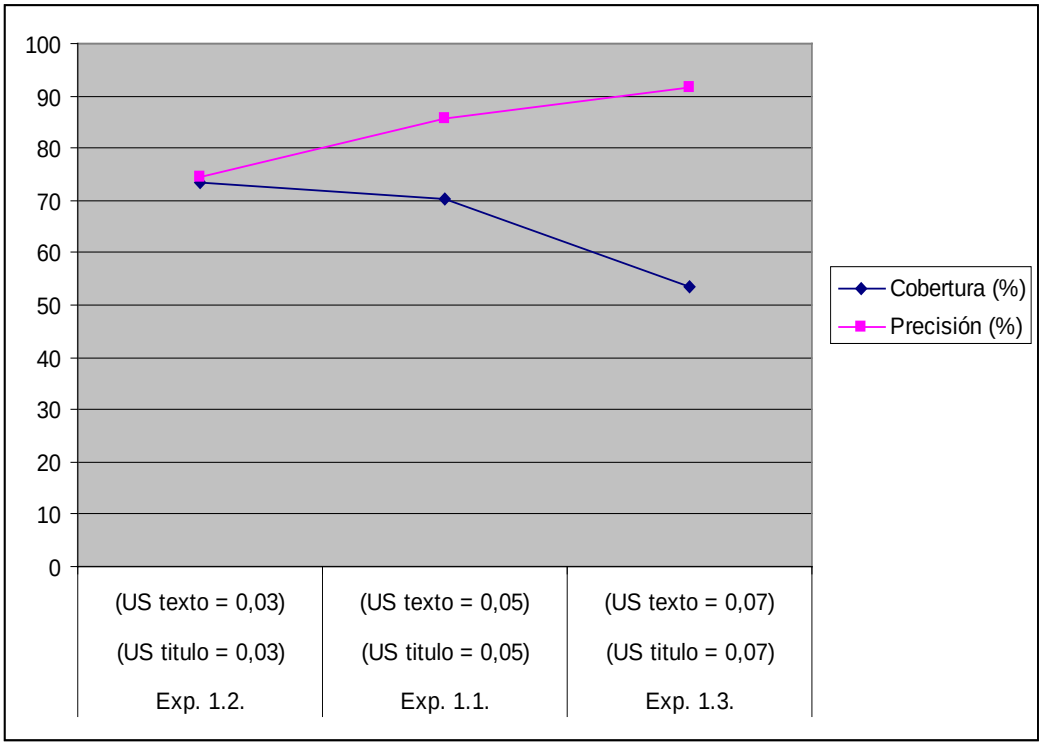


Figura 7.3: Relación entre precisión y cobertura en los experimentos cualitativos.

En la gráfica anterior se aprecia claramente como un aumento de la precisión produce una disminución de la cobertura y viceversa.

7.2.2. Experimentos cuantitativos

Seguidamente se muestran los dos experimentos cuantitativos que se han analizado para determinar los valores adecuados de los umbrales de similitud. Así como en los experimentos se trabajaba con las medidas de cobertura y precisión, en los experimentos cuantitativos se trabaja sobre todo con la medida F-Measure ya que es una combinación de las dos anteriores y resulta mucho más fácil comparar una sola medida. Esta F-Measure fue definida en el apartado 7.1 como:

$$F - Measure = \frac{1}{\frac{\alpha}{precisión} + \frac{(1-\alpha)}{cobertura}}$$

7.2.2.1. Experimento 2.1.

7.2.2.1.1. Generación de un corpus de evaluación

En este primer experimento cuantitativo se trabaja sobre el mismo corpus de noticias que en los experimentos cualitativos (Corpus del Experimento 1 contenido en el Apéndice A.1.). Corpus de 118 noticias extraído de la BD el 13 de octubre de 2008 a las 16:00. Lo que cambia en este caso son los umbrales de similitud ya que en vez de tratar únicamente 3 combinaciones diferentes de US de título y texto (0.3 y 0.3; 0.5 y 0.5; 0.7 y 0.7), se tratan más de 400 combinaciones.

Se incrementarán los umbrales de similitud de 0,01 en 0,01 yendo desde 0 (umbral en el cual se relacionarán todas las noticias) hasta 0,2. Se ha acotado el umbral en este rango ya que a partir de 0,2 los umbrales de similitud de título y texto no sólo no producen un aumento en la F-Measure sino que hacen que ésta disminuya enormemente.

Como grupos de referencia se han tomado los 11 mismos grupos especificados en 7.2.1.1.1.

7.2.2.1.2. Resultadosdel experimento

En este experimento se obtiene una primera aproximación a los valores óptimos de US(título) y US(texto) para el TRACKER. Al ser el primer experimento cuantitativo no se podrán extrapolar los valores obtenidos a otros corpus de noticias, pero los resultados extraídos servirán como referencia.

La siguiente gráfica tridimensional da una idea de los valores de US(titulo) y US(texto) para los que F-Measure es máxima. Se utiliza la medida F-Measure para $\alpha = 0{,}2$ ya que con este alfa se potencia la cobertura sobre la precisión en nuestro sistema (es preferible cubrir todos los temas aunque se incluyan noticias incorrectas en alguno de ellos, antes de que el sistema deje sin cubrir temas a costa de ser muy preciso en las relaciones).

Además de poder observar los máximos y mínimos, con esta gráfica también se puede apreciar la variación de los valores de F-Measure. También se determina en el suelo el contorno de valores referencia que pueden resultar útiles.

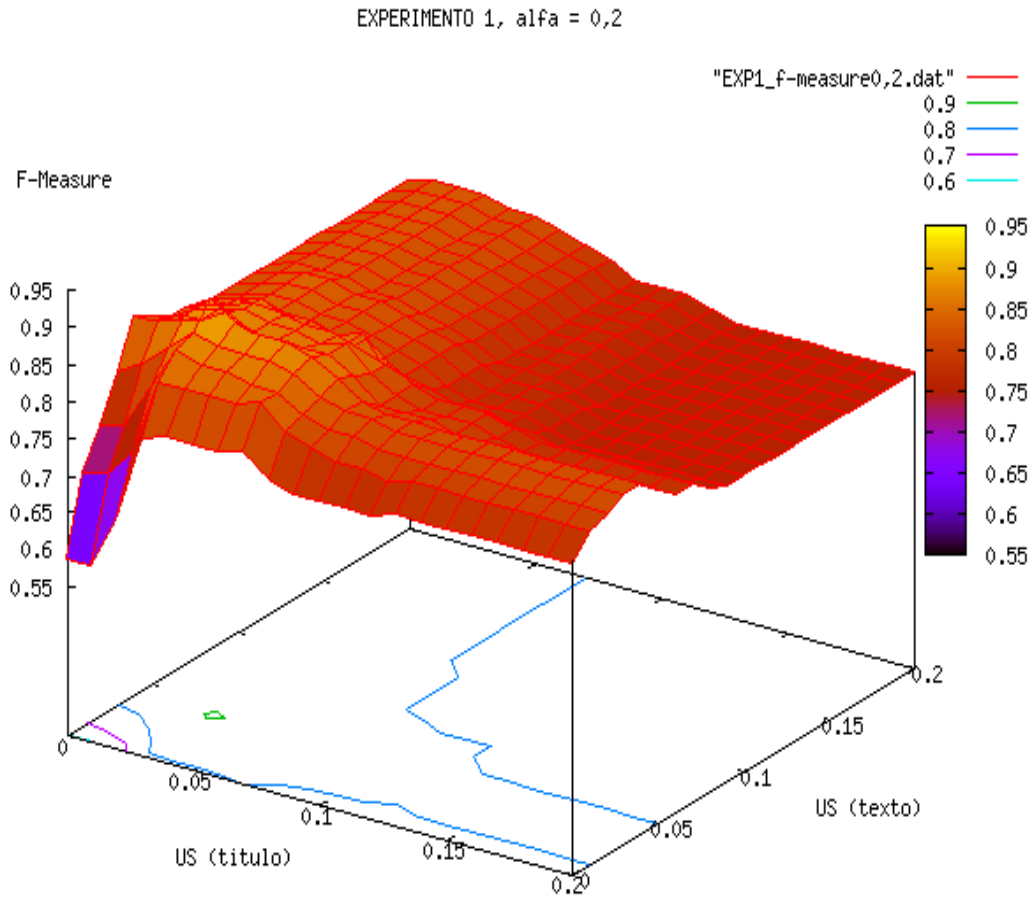


Figura 7.4: Gráfica tridimensional del Experimento 2.1.

Con las siguientes gráficas bidimensionales se completa la visión de la gráfica tridimensional ya que tratan los perfiles teniendo en cuenta la variación del US(texto) en primer lugar y la variación de US(título) en el segundo.

Valores de F-Measure representados por el US(título) en función de US(texto):

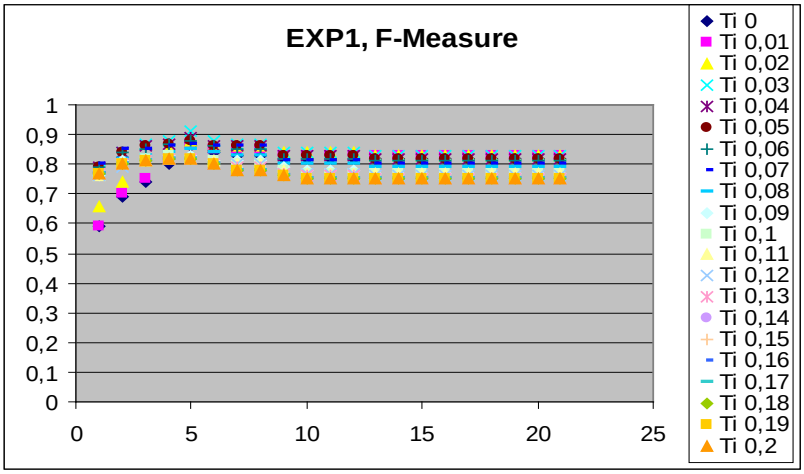


Figura 7.5: Gráfica de F-Measure en función de US(texto) para el Experimento 2.1.

Valores de F-Measure representados por el US(texto) en función de US(título)

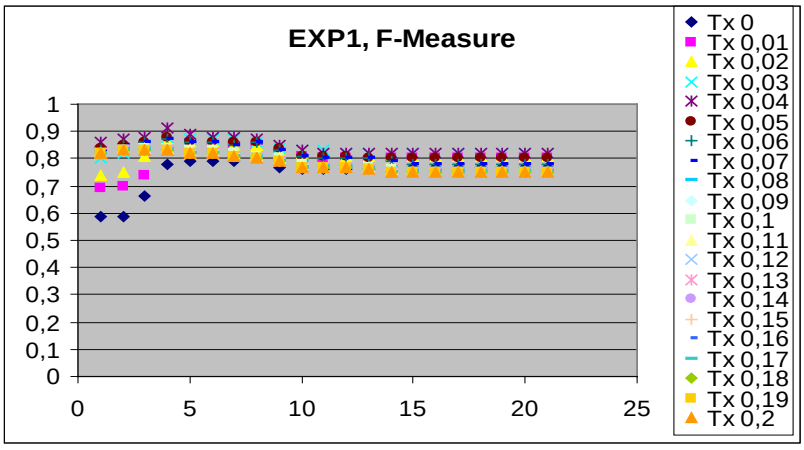


Figura 7.6: Gráfica de F-Measure en función de US(título) para el Experimento 2.1.

Se determina que los valores para los que se alcanza el máximo de F-Measure con $\alpha = 0{,}2$ son en el Experimento 2.1:

US(titulo) de 0,03 a 0,06 y US(texto) de 0,04

Estos valores de los umbrales de similitud quedan refrendados también por la medida F-Measure para $\alpha = 0{,}5$ como se muestra en las dos gráficas a continuación:

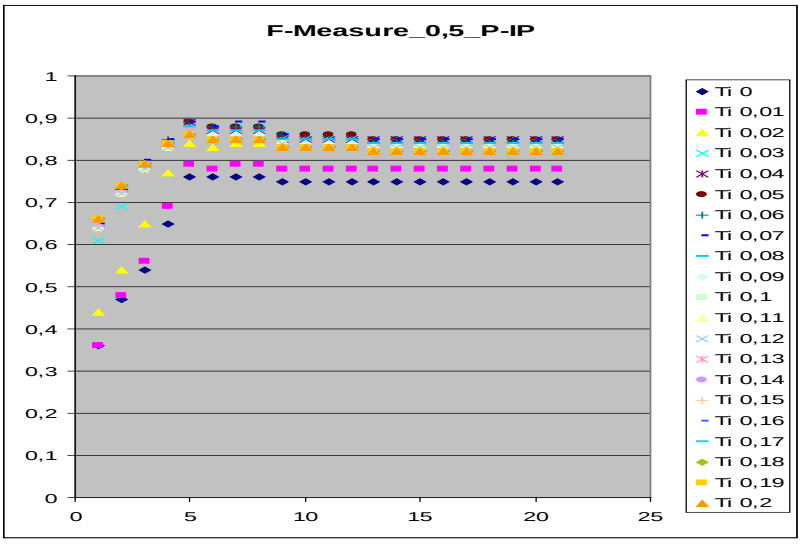


Figura 7.7: F-Measure con $\alpha = 0{,}5$ en función de US(texto) para el Experimento 2.1

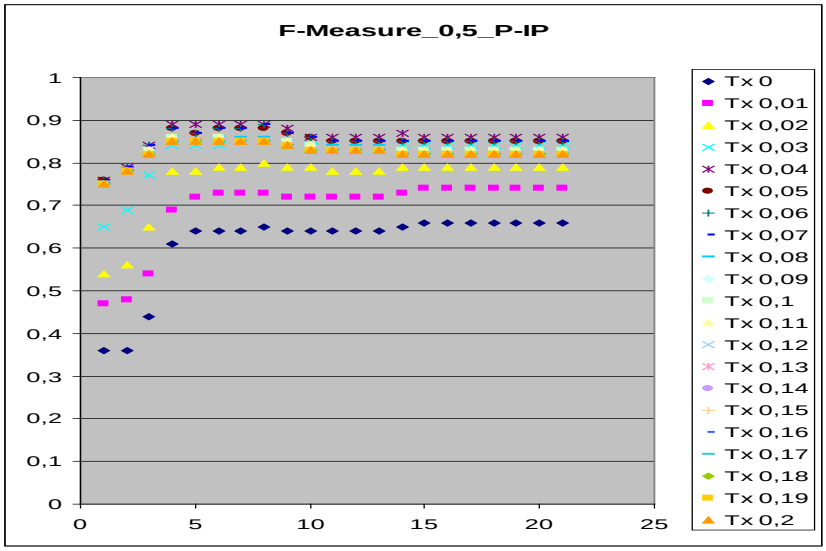


Figura 7.8: F-Measure con $\alpha = 0{,}5$ en función de US(texto) para el Experimento 2.1

7.2.2.1.3. Comparación de los resultados con las Baselines

BASELINES

	F-Measure $\alpha = 0{,}2$	**F-Measure** $\alpha = 0{,}5$
NADA	0,69	0,78
TODO	0,4	0,21
MEJOR RESULTADO TRACKER	0,91	0,89

Por tanto, con el sistema TRACKER se consiguen unas mejoras respecto a las baselines de:

- Mejora de entre el 14% y el 32 % respecto a no agrupar ninguna noticia.

- Mejora de entre el 127% y el 324% respecto a agrupar todas las noticias en un solo tema.

7.2.2.2. Experimento 2.2.

Este segundo experimento cuantitativo supone la ejecución del TRACKER a partir de las 6 horas del 7 de julio de 2008 sobre el conjunto de post descargados hasta ese momento en la base de datos.

En total se trabaja sobre un corpus de 104 noticias (detalladas en el Anexo A.2.)

7.2.2.2.1. Generación de un corpus de evaluación

Para este corpus de noticias se han identificado 17 temas comunes o clusters que contienen más de una noticia.
Los 17 clusters contienen 58 noticias.
Los temas son los siguientes:

CLUSTER	TEMA	Nº de noticias asociadas al tema
1	Final de Wimbledon. Victoria de Nadal	8
2	Rescate de Ingrid Betancourt	3
3	Muertes en la carretera en el fin de semana	3
4	Festival Rock in Rio	2
5	Crisis financiera	6
6	Niegan la entrada en el Ejército a un transexual	2
7	Moción de censura contra Laporta	3
8	El Coloso ¿De Goya?	2
9	Lluvias en el noreste de España	2
10	Juegos Olímpicos	2
11	Atentado en Kabul	2
12	F1. Gran Premio de Silverstone	4
13	Sobre el aborto	2
14	Manifiesto en defensa de la Lengua Común	2
15	Reuniones en el seno del PP	5

	de Cataluña	
16	Congreso del PSOE, nuevas líneas de actuación del partido	8
17	Situación actual de Telecinco	2

7.2.2.2.2. Resultados del experimento

Los datos de la tabla de F- Measure para $\alpha = 0{,}2$, que son los más útiles para nuestra aplicación, se muestran gráficamente a continuación:

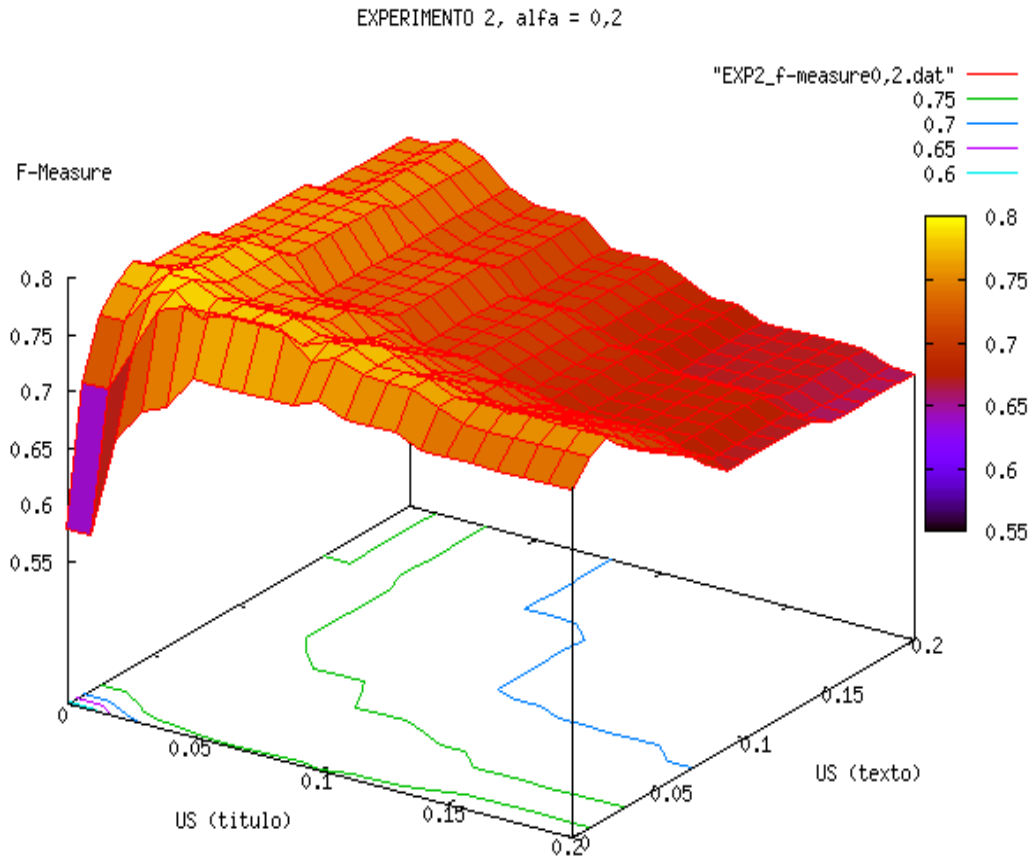


Figura 7.9: Gráfica tridimensional del Experimento 2.2.

Donde se observa que en este experimento ya no hay un máximo tan claro como en el experimento 2.1. sino que los valores altos de F-Measure oscilan a lo largo de un rango más amplio de valores de US(título). El otro umbral de similitud, US(texto), permanece más o menos estable en los máximos como en el experimento anterior.

Por otra parte también vemos que los valores de F-Measure han disminuído sensiblemente. La razón de esta disminución es difícil de determinar puesto que aunque son corpus de noticias parecidos en cuanto a tamaño y número de clusters en ambos experimentos, las noticias de cada corpus son muy diferentes. En cualquier caso, la

disminución de F-Measure en este experimento no resulta preocupante ya que en el proyecto TRACKER no se busca maximizar su valor sino encontrar los valores de los umbrales de similitud que consiguen ese máximo.
Valores de F-Measure representados por el US(título) en función de US(texto):

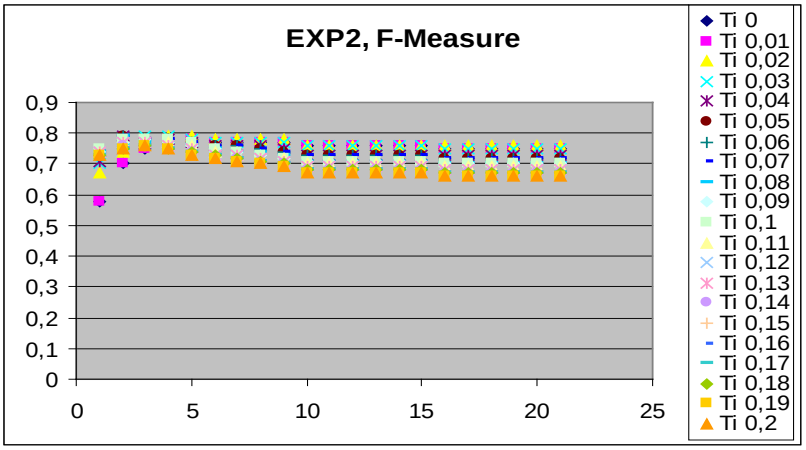


Figura 7.10: F-Measure en función de US(texto) para el Experimento 2.2.

Valores de F-Measure representados por el US(texto) en función de US(título):

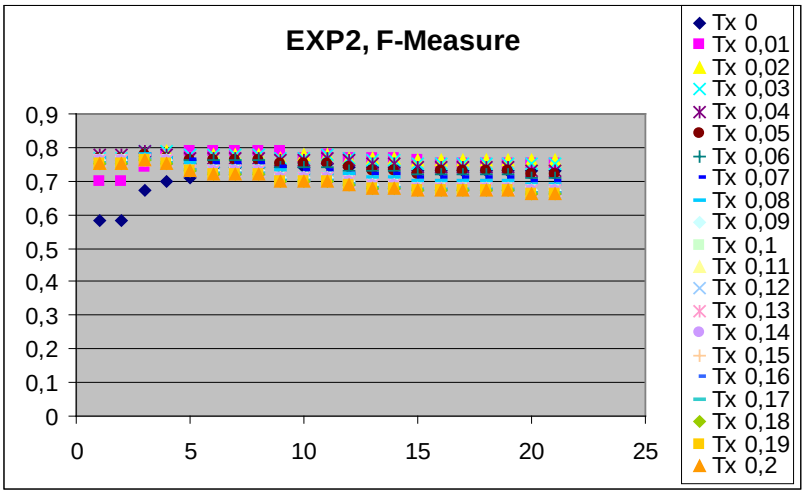


Figura 7.11: F-Measure función de US(título) para el Experimento 2.2.

En el Experimento 2.2, se determina que los valores para los que se alcanza el máximo de F-Measure con $\alpha = 0{,}2$ son:

US(titulo) de 0,02 a 0,08 y US(texto) de 0,01 a 0,04

Estos valores de los umbrales de similitud quedan refrendados también por la medida F-Measure para $\alpha = 0{,}5$ como se muestra en las dos gráficas a continuación:

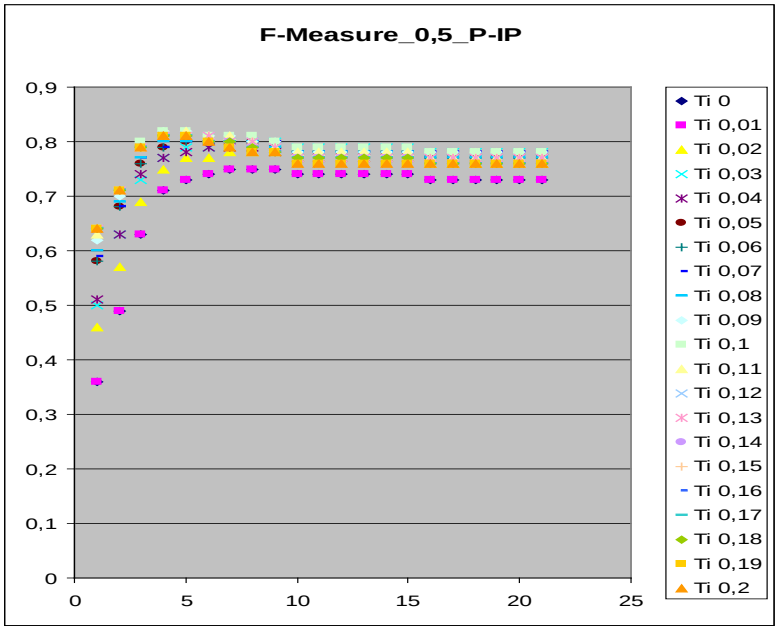


Figura 7.12: F-Measure con $\alpha = 0{,}5$ en función de US(texto) para el Experimento 2.2.

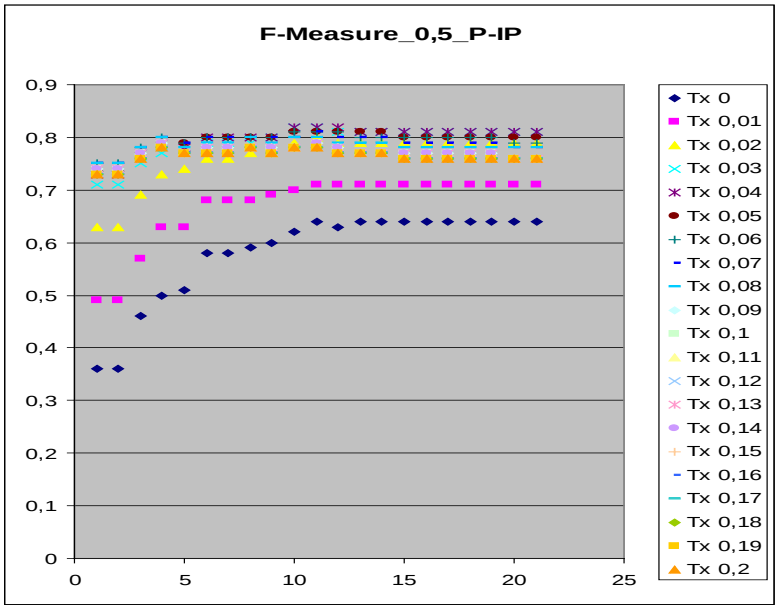


Figura 7.13: F-Measure con $\alpha = 0{,}5$ en función de US(título) para el Experimento 2.2.

7.2.2.2.3. Comparación de los resultados con las Baselines

BASELINES

	F-Measure $\alpha = 0{,}2$	**F-Measure** $\alpha = 0{,}5$
NADA	0,66	0,75
TODO	0,29	0,14
MEJOR RESULTADO TRACKER	0,79	0,82

Por tanto, con el sistema TRACKER se consiguen unas mejoras respecto a las baselines de:

- Mejora de entre el 9% y el 20 % respecto a no agrupar ninguna noticia.

- Mejora de entre el 172% y el 485% respecto a agrupar todas las noticias en un solo tema.

7.2.2.3. Experimento 2.3.

En este último experimento cuantitativo se ejecuta la aplicación TRACKER a partir de las 22 horas del 15 de junio de 2008 sobre el conjunto de post descargados hasta ese momento en la base de datos.

En total se trabaja sobre un corpus de 97 noticias (detalladas en el Anexo A.3.)

7.2.2.3.1. Generación de un corpus de evaluación

Se han identificado 14 temas comunes o clusters que contienen más de una noticia.
Los 14 clusters contienen 47 noticias.
Estos clusters son los siguientes:

CLUSTER	TEMA	Nº de noticias asociadas al tema
1	Jornada laboral de 65 horas	5
2	Huelga de transportistas	4
3	Directiva Europea sobre Temps de Treball	3
4	Expo Zaragoza 2008	2
5	Calentamiento global	2
6	Esperanza Aguirre en la Asamblea de Madrid	7
7	El PP en el País Vasco	3
8	Sobre Sevilla	3
9	Día del Orgullo Gay	2
10	Explotación infantil	2
11	Sobre un pueblo llamado Torrentino	2
12	Machismo	6
13	Barack Obama	3
14	Sobre el catalán	3

7.2.2.3.2. Resultados del experimento

El rango de valores de umbral de similitud es mayor para este experimento (se prolongan más los valores de la variable US(tiítulo)) debido a que los máximos de F-

Measure están desplazados hacia valores más altos de US(título) que en los experimentos anteriores.

Para ilustrar gráficamente la situación se expone la siguiente gráfica tridimensional:

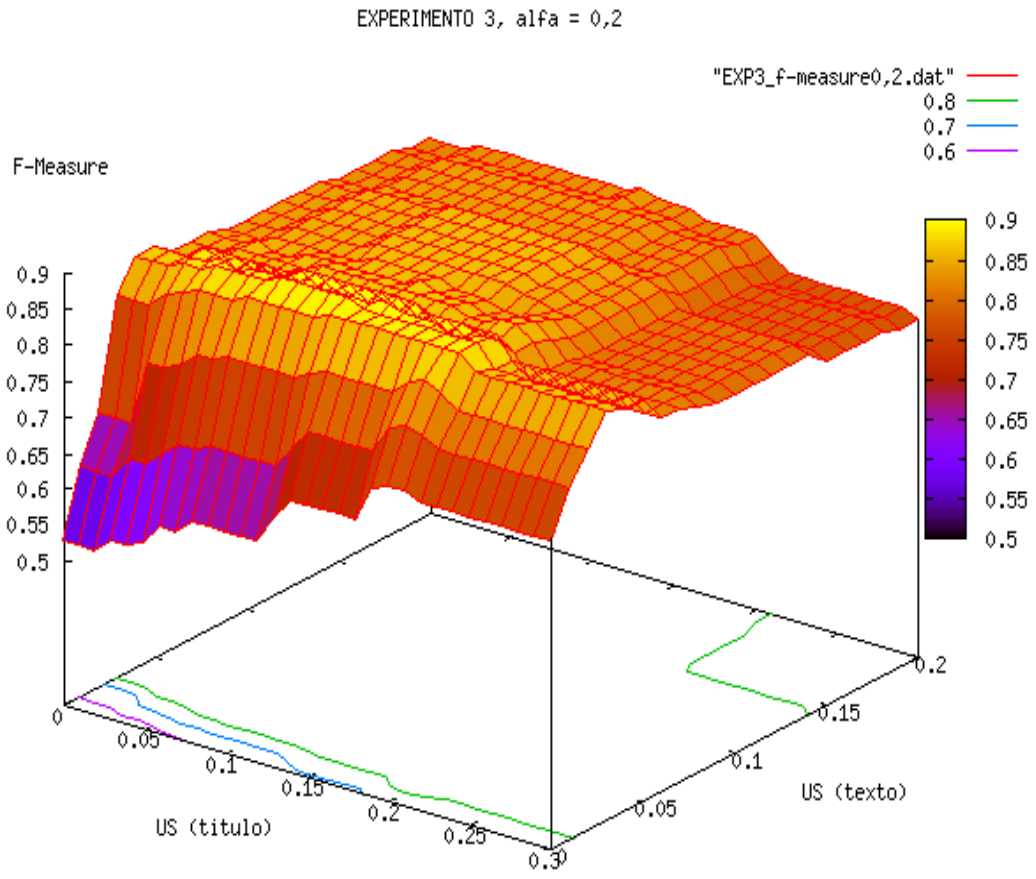


Figura 7.14: Gráfica tridimensional del Experimento 2.3.

Observamos como en este experimento la variación de F-Measure es menor aún que en el experimento anterior. No hay grandes variaciones (salvo para valores muy extremos) y hay muchos valores de US(título) que maximizan el valor de la medida de evaluación. Sin embargo, los valores de US(texto) que maximizan el valor de F-Measure siguen siendo aproximadamente los mismos.

Estas observaciones se corroboran con las gráficas bidimensionales que muestra el perfil de la anterior.
Valores de F-Measure representados por el US(título) en función de US(texto):

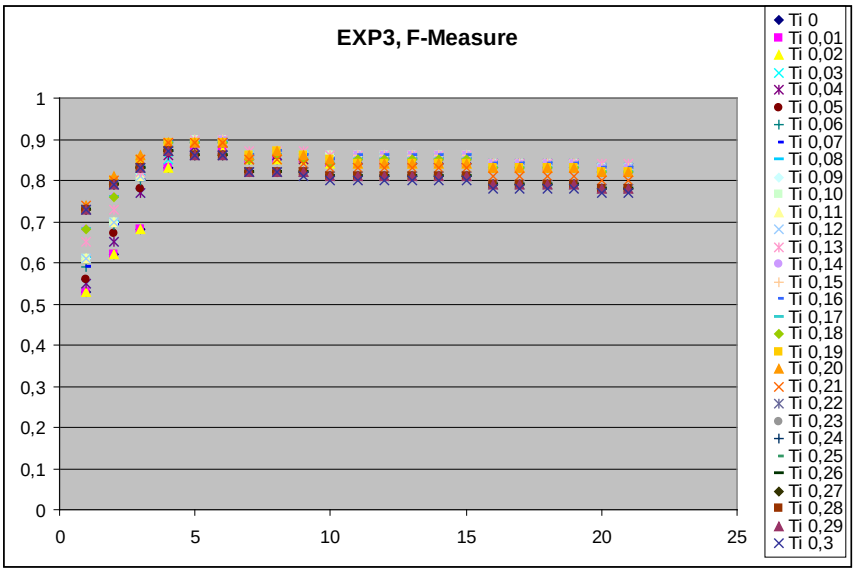


Figura 7.15: F-Measure en función de US(texto) para el Experimento 2.3.

Valores de F-Measure representados por el US(texto) en función de US(título):

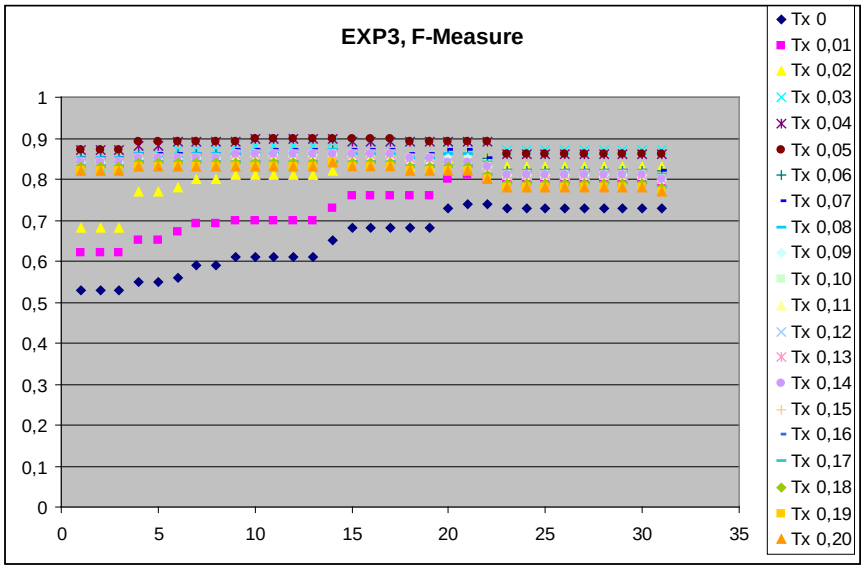


Figura 7.16: F-Measure en función de US(título) para el Experimento 2.3.

En el Experimento 2.3, se determina que los valores para los que se alcanza el máximo de F-Measure con $\alpha = 0{,}2$ son:

US(titulo) de 0,09 a 0,16 y US(texto) de 0,04 a 0,05

Y como en los experimentos cuantitativos anteriores, los valores determinados de US(texto) y US(título) se ven refrendados también si los evaluamos mediante F-Measure con $\alpha = 0{,}5$ en vez de $\alpha = 0{,}2$. A continuación se muestran estas tablas:

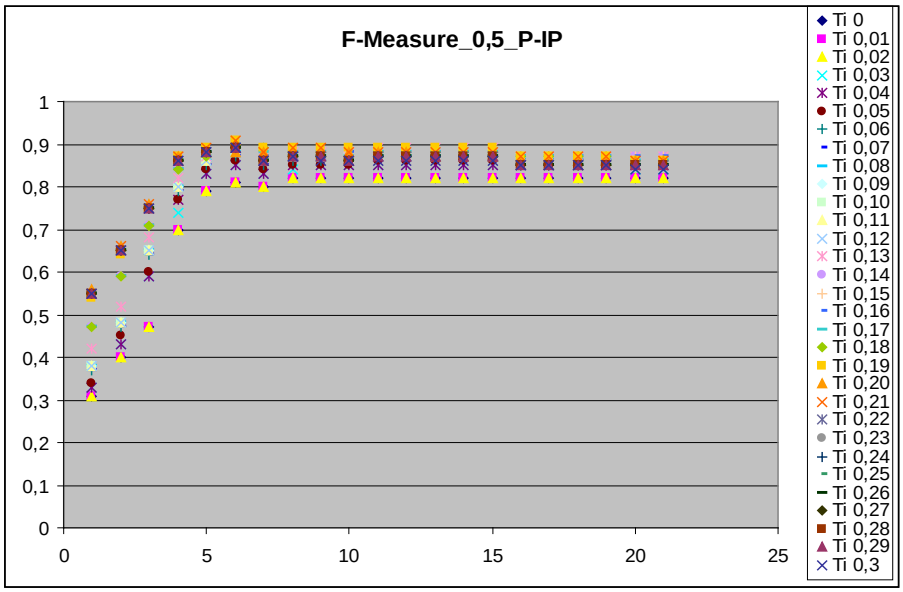


Figura 7.17: F-Measure con $\alpha = 0{,}5$ en función de US(texto) para el Experimento 2.3.

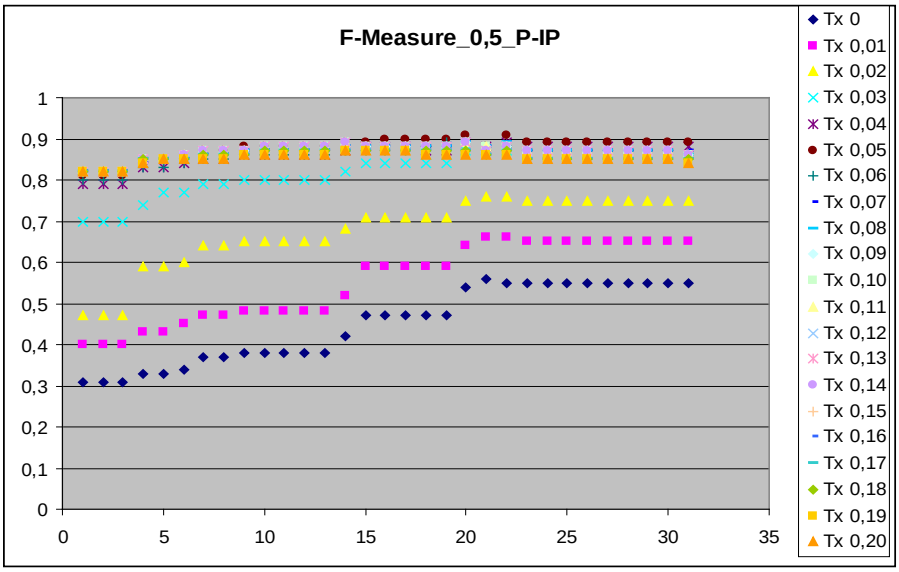


Figura 7.18: F-Measure con $\alpha = 0{,}5$ en función de US(título) para el Experimento 2.3.

7.2.2.3.3. Comparación de los resultados con las Baselines

BASELINES

	F-Measure $\alpha = 0{,}2$	**F-Measure** $\alpha = 0{,}5$
NADA	0,71	0,8
TODO	0,28	0,13
MEJOR RESULTADO TRACKER	0,9	0,91

Por tanto, con el sistema TRACKER se consiguen unas mejoras respecto a las baselines de:

- Mejora de entre el 13% y el 26 % respecto a no agrupar ninguna noticia.

- Mejora de entre el 221% y el 600% respecto a agrupar todas las noticias en un solo tema.

Conclusiones de los experimentos cuantitativos

La variable de umbral de similitud de texto es la que produce mayor dispersión de los datos, por tanto es la más importante de ajustar. Para los tres experimentos se obtiene que para un valor de US (texto) = 0,04 se encuentra el máximo de la medida F-Measure. Aunque también para un valor de US (texto) = 0,01 – 0,05 se han encontrado máximos en los experimentos. En cualquier caso la oscilación es muy pequeña (un 4%).

Por otro lado la variable de umbral de similitud de título es menos importante porque una variación en sus valores no produce una gran variación de los valores de F-Measure. Los valores que producen el máximo son para US (titulo) = 0,02 – 0,16. Se ve que hay una oscilación mayor para los valores óptimos de este umbral (14%). De todas formas es menos importante ajustar bien esta variable.

Se observa que los datos de F-Measure del EXPERIMENTO3 son más homogéneos.

	US (título)	**US (texto)**
Experimento 2.1.	0,03 - 0,06	0,03 - 0,05
Experimento 2.2.	0,02 - 0,08	0,01 - 0,04
Experimento 2.3.	0,09 - 0,16	0,04 - 0,05

Por la tabla anterior se aprecia claramente que para el umbral de similitud de texto existe un valor para el cual se obtiene el máximo rendimiento del TRACKER. Este

valor es 0,04. En el umbral de similitud de título no se puede establecer ningún umbral que maximice los tres experimentos, pero si hay valores que ofrecen un altísimo rendimiento en todos los casos.

7.3. Discusión sobre los experimentos

De los experimentos realizados sobre la aplicación se extraen algunos puntos

No existen unos valores fijos de US (título) y US (texto) que aseguren los mejores resultados de relación de noticias para cualquier corpus de noticias. Hay que recordar que el proceso de relación de noticias tiene un alto componente subjetivo por lo que un cambio en el criterio de relación cambiará la evaluación del trabajo del TRACKER.

- Pese a lo anterior se puede establecer que para las noticias de la base de datos Politiktracker resulta eficiente aplicar unos valores de **US (título)** que oscilan **entre 0,02 y 0,11** y unos valores de **US (texto)** de entre **0,02 y 0,04.**

- Aunque no existen valores de los US que aseguren optimizar el rendimiento del tracker, se puede decir que los valores de US (texto) y US (título) deben mantener una proporción equilibrada de sus valores. Grandes diferencias entre ambos aseguran que no se obtengan buenos resultados de tracking.

- Los US utilizados también dependerán en gran medida del corpus de datos con el que se esté tratando. Si es un corpus con un elevado número de temas, los valores idóneos de US se verán incrementados. Si el corpus tiene un reducido número de temas, los mejores valores de US serán mucho más pequeños.

- En cualquier caso, para nuestra aplicación resulta más interesante establecer US pequeños para así aumentar la cobertura media del tracker (cubriendo el mayor número de temas). Es preferible fallar en algunas relaciones que dejar temas sin cubrir. Si se fallan relaciones, el usuario rápidamente identifica el fallo con echar un vistazo a la noticia; sin embargo, si se deja sin cubrir algún tema resulta muy costoso para el usuario identificar todas las noticias de ese tema.

- Teniendo en cuenta el punto anterior y las medidas de evaluación utilizadas en el apartado de experimentación, hay que determinar que resulta mucho más interesante utilizar la medida F-Measure_0,2 antes que la medida F-Measure_0,5 ya que el valor $\boldsymbol{\alpha = 0,2}$, y ese valor potencia los valores de cobertura en lugar de los de precisión. El valor $\boldsymbol{\alpha = 0,5}$ da un peso más igualado a la precisión.

- Por las gráficas del punto 7.2.2 se puede establecer que el US (texto) produce una mayor variación de la medida F-Measure que el US (título), esto se ve reflejado en la dispersión de los puntos de las gráficas. Esta característica refleja el mayor peso del campo texto frente al campo título, y se debe principalmente a la mayor longitud del campo texto lo cual supone un mayor rango de similitudes entre las noticias por el campo texto.

- Al tener mayor peso el umbral de similitud de texto, US(texto), sobre el buen funcionamiento del TRACKER, son muy satisfactorios los resultados obtenidos en los experimentos porque aunque no se han encontrado valores universales (dentro de los experimentos) de US(título), si se han encontrado para US(texto).

- Se debe tomar como valor adecuado y punto de partida para otros experimentos un US(texto) = 0,04.

Además de las conclusiones numéricas formuladas a partir de los resultados de los experimentos cuantitativos, también se sacan algunas conclusiones del funcionamiento del TRACKER en cuanto a calidad de funcionamiento. Al tratarse la aplicación de detección y seguimiento de noticias, la calidad de la misma podrá establecerse a partir de la forma de unir noticias correcta o incorrectamente. Las conclusiones sobre la calidad del sistema son:

- Los errores en la relación de noticias casi siempre se producen por nombres propios (o en algunos casos comunes) que se encuentran fuera de contexto. Como ejemplo se puede destacar la confusión que producen nombres como “Zapatero” o “Rajoy” a la aplicación. Estas palabras se utilizan muchas veces en noticias que poco tienen que ver con los protagonistas pero que automáticamente son relacionadas por el TRACKER con otras noticias por error.

- Otra de las grandes conclusiones que se sacan sobre las relaciones de las noticias, es el alto grado de subjetivismo y dificultad que conlleva el relacionar noticias o hechos para una persona (no digamos ya para un sistema de TDT). Establecer los límites de un suceso resulta complicado, y así, un hecho como pueden ser las elecciones de un país puede contener muchos subtemas en forma de noticias, y a su vez estar contenido en otros temas más generales.

Capítulo 8

Conclusiones

A lo largo de este trabajo se han desarrollado las herramientas necesarias para el tratamiento de las noticias de una base de datos. No sólo se han tratado las noticias relacionándolas entre sí para la formación de grupos temáticos sino que también se ha realizado un mecanismo de evaluación de la aplicación. Es decir, se ha completado el funcionamiento del TRACKER de detección y seguimiento de noticias tal y como se especificó en la fase de análisis y diseño de la aplicación. Se han satisfecho los requisitos establecidos y se han extraído datos que permitirán mejorar en un futuro el funcionamiento del método y del programa.

Por tanto, teniendo en cuenta lo expuesto en el párrafo anterior, se deben extraer conclusiones sobre el proceso de creación del TRACKER. Desde el momento en que empezaron las conversaciones con el tutor para sentar las bases del proyecto (hace ya bastantes meses) hasta su puesta en funcionamiento cumpliendo los requisitos formulados y la elaboración de la documentación y la memoria.

Todo el trabajo realizado durante el último año me ha permitido comprender lo que significa el desarrollo de un proyecto de cierta magnitud. Me parece indispensable la realización de un trabajo de estas características para completar el aprendizaje de la carrera, ya que con los trabajos a menor escala que se realizan en las asignaturas no se consigue tener una visión global de lo que es un proyecto.

Las conclusiones generales que extraigo de la realización del PFC son:

- La importancia de la **investigación** en el proceso de desarrollo. Gracias a la búsqueda y estudio de la información relacionada con el proyecto he conseguido afianzar los conceptos que posteriormente trataba de desarrollar en el programa. Este fue un trabajo duro y poco agradecido puesto que no se plasmaba en resultados palpables, pero ahora con la perspectiva que me ha dado el paso del tiempo veo que en las tempranas fases de investigación son vitales para conseguir resultados.

- La importancia de un buen **análisis** y **diseño**. Para mi trabajo ha sido importantísimo este punto, puesto que sin un buen diseño el acoplamiento de los módulos de la aplicación hubiera sido difícil. Me ha resultado complicado realizar un buen análisis y diseño de la aplicación ya que durante la carrera se tiene a codificar las aplicaciones a contrarreloj y más instintivamente. Ahora, viendo los resultados de un diseño correcto se entiende el ahorro de tiempo que supone.

- Durante la fase de desarrollo se han intentado seguir buenas prácticas de programación siguiendo las pautas que marcan **metodologías de desarrollo de software**. Este punto es importante no sólo para el presente trabajo sino también para posteriores desarrollos del TRACKER que se basen en él. Además al utilizar un lenguaje de programación desconocido, he buscado aprender a manejar lo más correctamente posible la **documentación** y **estructuración** del código.

- He aprendido la importancia de realizar una buena batería de **pruebas**. Si bien ya conocía la importancia de realizar bien la fase de pruebas, con un proyecto como éste tan basado en los resultados numéricos, se pone de manifiesto que para mejorar la eficiencia de la aplicación este punto es básico. Y no sólo fue importante la batería de pruebas final, ya que en las fases tempranas de desarrollo de la aplicación también fueron indispensables las pruebas modulares realizadas.

- Me gustaría reseñar la importancia de la **comunicación con el equipo**. En este proyecto que forma parte de un proyecto más general como MEMETRACKER, es básica la comunicación entre los miembros del equipo. Para mí, que partía de cero en muchos aspectos, el diálogo con el resto de personas del equipo ha sido muy enriquecedor. Además considero muy útil el haber realizado un proyecto con varias personas para acostumbrarme a la manera de trabajar en las empresas donde la comunicación dentro de los equipos de trabajo es importantísima.

También quisiera reseñar los siguientes puntos de interés en la realización del proyecto:

- La **orientación a objetos**. Que ha sido el eje central de desarrollo, tanto para la implementación utilizando JAVA como con la utilización de UML para modelar el sistema en la fase de análisis y diseño. Es otra filosofía de desarrollo que me resulta de particular interés ya que durante la carrera no he tenido la oportunidad de conocerla. Me parece que debería ser obligatorio su aprendizaje para un ingeniero técnico informático dada su importancia en el mundo de la informática hoy en día.

- La utilización de un **entorno de desarrollo colaborativo**. En mi caso la utilización del IDE Eclipse me ha permitido compartir código con el resto de componentes del proyecto y también mantener una línea de desarrollo más centrada en el objetivo.

- Utilización de una Wiki en la que compartir e ir explicando los textos, esquemas y desarrollos del proyecto, y de esta manera entender cada una de las partes que realizaban otras personas involucradas.

Para terminar hay que concluir que se consideran satisfactoriamente logrados los objetivos del proyecto en cuanto a eficacia y eficiencia, habiendo completado y justificado experimentalmente un estudio de relaciones de noticias, y habiendo aportado un marco genérico para la aplicación de TDT a otros contextos y entornos. Se han evaluado los procesos de relación de noticias dentro de la aplicación constatando que hay unos valores para los cuales el algoritmo elaborado funciona muy eficazmente.

También se han sentado las bases para futuras mejoras en el rendimiento de la aplicación y adaptación de otros módulos.

Capítulo 9

Líneas futuras

Este apartado recoge algunas propuestas para futuras aportaciones al estudio de relación de noticias y de la detección y seguimiento de tópicos. También se establecen posibles mejoras para el sistema informático implementado en este proyecto.

- Corroborar las conclusiones de este proyecto con estudios similares pero con distintos corpus de noticias en cuanto a tamaño y grupos de temas se refiere. Sería muy conveniente también estudiar el comportamiento de la aplicación TRACKER desde el punto de vista del origen de los datos. Cada periódico digital o blog de la WWW tiene unas características particulares.

- Tratar el proceso de relación de noticias más en profundidad, haciendo hincapié en los parámetros modificables del módulo analizador.

Aunque la unidad de trabajo del tracker es en nuestro caso las noticias políticas sacadas de una base datos (Politiktracker), se pueden sacar varias conclusiones que afectan a futuros desarrollos y mejoras de la aplicación. La modularidad con que ha sido diseñada la aplicación permite muchos cambios que no supondrán un gran consumo de tiempo y recursos. Por tanto, sobre el objeto con el que trabaja nuestro relacionador de noticias se pueden establecer las siguientes líneas futuras:

- En nuestro caso se extraen unidades de información (noticias) de una BD que es común a todo el macroproyecto MEMETRACKER. Se puede adaptar el desarrollo realizado para extraer la información de otras fuentes como podrían ser otras BD diferentes, archivos de texto u otros almacenes en los que se encuentre información interesante.

- Aunque el proyecto se centra en el análisis de noticias políticas, por ser éstas de especial interés y relevancia en los medios digitales, también se puede ejecutar la aplicación sobre un almacén de datos que contenga noticias de otra índole. Para este caso es importante determinar las características principales del vocabulario extraído para mejorar el funcionamiento del tracker.

- El idioma con el que trabaja el tracker es el castellano por ser el utilizado en los medios digitales que contienen las noticias tratadas. Sería, como se explica en el apartado del Estado del Arte y diseño del proyecto, muy sencillo adaptar el funcionamiento de la aplicación a un corpus de datos que esté contenido en medios extranjeros y por tanto en otros idiomas. Esto no reduciría de manera alguna el rendimiento de la aplicación.

- Por último, respecto al medio con el cual se trabaja, se puede concluir que el tratamiento de datos que estén en otro formato diferente al escrito sí conllevaría grandes cambios en la aplicación. Es necesario concluir esto ya que actualmente las fuentes de información en Internet contienen en muchos casos archivos de audio y de video que nuestra aplicación no está tratando actualmente.

Respecto al desarrollo de la aplicación y las herramientas utilizadas, se pueden establecer otras líneas futuras que marcan el camino a seguir en posteriores desarrollos. El módulo analizador de la aplicación representa el núcleo y es el analizador el que permite relacionar las noticias de la manera en que se hace. La información escrita tratada tiene numerosas características que la definen y permiten relacionarla entre sí. Tal y como se explicó en el apartado 4.6.1.1 de esta memoria, existen numerosas expresiones que permitirían mejorar el módulo analizador y el funcionamiento del tracker elevando los indicadores de efectividad.

- Las expresiones temporales son utilizadas muy frecuentemente en aplicaciones de TDT. No significa sólo acudir a la fecha de publicación de la noticia como es el caso de nuestra aplicación (aunque esto también proporciona información valiosa) sino sobre todo acudir a los vocablos utilizados que tienen algún componente temporal. Resultados sacados de otras investigaciones indican aumentos sustanciales en el rendimiento de aplicaciones de TDT y de RI en general.

- Expresiones de lugar, nombres propios y otras características particulares de los textos también permitirían aumentar el rendimiento de la aplicación. En nuestra aplicación sólo se tratan estas expresiones de forma general sin utilizar ninguna base de datos o almacén con grupos de términos que relacionen una noticia con una región, una persona o una entidad específica.

Tratando ya específicamente la relación entre las noticias y los grupos de semejanza creados se definen las siguientes líneas futuras.

- Actualmente no se extrae ninguna información representativa de cada grupo de noticias, esto es lógico pues no se requería extracción de la información sino sólo su seguimiento y relación, pero con los mecanismos de la aplicación actual no se establece una noticia que represente al grupo de noticias. Esto sería muy útil para hacerse a la idea de la temática del grupo y no requeriría extracción de la información ni de atributos comunes a las noticias del grupo.

- Se da el caso de que en nuestra aplicación dos noticias se relacionan si están relacionadas en un sentido u otro, esto es, cuando el tracker establece diferentes relaciones como A → B, B → A ó A ←→ B donde A y B son dos de las noticias de la base de datos, le está dando la misma importancia a las tres relaciones cuando lógicamente la relación A ←→ B tendrá un peso mayor que las otras dos. Sería interesante establecer un peso específico a cada enlace entre noticias.

- También respecto a los enlaces de noticias dentro de un grupo, cabe reseñar que la aplicación de tracking no tiene en cuenta el número de enlaces que hay en el grupo. Por supuesto cuantos más enlaces haya en un grupo, más cohesión habrá entre las noticias de ese grupo y más certeza existirá de que esas noticias son de temática común.

- Sobre las noticias mal relacionadas que se muestran en los experimentos cualitativos del apartado 7.2.1, hay que resaltar la importancia que tiene la identificación de los fallos que ha cometido el tracker. Actualmente sólo es posible con el análisis detallado de cada noticia, pero sería posible identificar las palabras que producen confusión a la aplicación gracias al módulo evaluador WePs. Con este módulo se pueden identificar las noticias mal relacionadas y tras esta identificación sería posible deducir los vocablos que han propiciado esa relación. Tras la identificación de esos vocablos deberían ser tratados de diferente manera por el módulo analizador para próximas noticias.

- Hay que recordar que el tracker utiliza índices para relacionar noticias, y también que noticias que superan una cierta antigüedad son eliminadas del índice. Esto resulta necesario para producir búsquedas eficientes en cuanto a tiempo y también para producir un ahorro de espacio de almacenamiento. Sin embargo sería interesante analizar cual es límite de tiempo más adecuado para eliminar noticias del índice. No sólo eso, sería muy útil implementar mecanismos que permitieran dejar en el índice noticias sobre temas que tuvieran visos de repetirse en el futuro (p.ej. en noticias sobre secuestros).

- Analizar otros campos de la noticia como el autor de la noticia que podrían proporcionar datos debido a la particular forma de escribir que tiene cada persona.

- Introducir expresiones temporales, de lugar, nombres propios y expresiones idiomáticas a la aplicación; introduciendo éstas en una base de datos y accediendo a la BD para mejorar la eficiencia del TRACKER [9].

- Mejorar también el sistema de detección de sucesos aplicando sinónimos de palabras.

- Relacionado con el punto anterior, establecer mecanismos de aprendizaje automático para el tracker de manera que las relaciones mal establecidas no se vuelvan a establecer en el futuro.

- Analizar y relacionar noticias con contenidos audiovisuales.

Capítulo 10

Bibliografía y otros recursos

[1] http://www.nist.gov/speech/tests/tdt/

[2] Bruce Eckel. “Piensa En Java”. McGraw-Hill. 2007.

[3] Jesús Sánchez Allende. “Java 2 : iniciación y referencia**”.** McGraw-Hill / InterAmericana de España. 2005.

[4] Jim Arlow, Ila Neustadt. “UML 2 And The Unified Process: Practical Object-Oriented Analysis And Design”. Pearson. 2005.

[5] Ricardo Baeza-Yates, B.Ribeiro-Neto. “Modern Information Retrieval”. Addison Wesley Longman. 1999.

[6] Erik Hatcher, Otis Gospodnetić. “Lucene in Action”. Manning. 2004.

[7] Charles L. Wayne. “Multilingual Topic Detection and Tracking:

Successful Research Enabled by Corpora and Evaluation". Ft. Meade, MD 20755-6514.
2005.

[8] Canhui Wang, Min Zhang, Shaoping Ma, Liyun Ru. "Automatic Online News Issue Construction in Web Environment". State Key Lab of Intelligent technology & systems, Tsinghua National Laboratory for Information Science and Technology, CS&T Department, Tsinghua University, Beijing, 100084, China P.R.
2008.

[9] Juha Makkonen, Helena Ahonen-Myka, Marko Salmenkivi. "Simple Semantics in Topic Detection and Tracking". Department of Computer Science, P.O. Box 26 (Teollisuuskatu 23), FIN-00014, University of Helsinki, Finland.
2003.

[10] Módulo de evaluación: Weps People Search Task (WePS) Evaluation Workshop. NLP & IR Group of UNED (Madrid). Javier Artiles. 2008.

Apéndices

A. Colección de noticias de los experimentos

A.1. Corpus del Experimento 1

El corpus con el que se trabaja en los experimentos 1.1, 1.2, 1.3 y 2.1 se detalla a continuación. El campo ID POST es el que identifica a cada post de manera unívoca en la base de datos, el TITULAR sirve para dar una idea del tema o tópico al que se refiere la noticia, y finalmente en el campo CLUSTER con el que se indica si el post está asociado a algún tema que contenga más de una noticia.

Si el campo del cluster está vacío significa que el post representa en sí mismo un grupo temático con una sola noticia.

ID POST	TITULAR	CLUSTER
40950	Castro votaría por Husein Obama	1
40951	John versus Husein	1
40968	ATRAPADOS EN LA CRISIS	8
40997	El Gobierno acelera la aplicación del Plan de Adaptación de los quitamiedos;	
40998	El PSOE asegura que los presupuestos de Interior permitirán profundizar en la política	

	de	
40999	González ha calificado de buena estrella penitenciaria el veloz acceso	
41000	Contenido de la propuesta de Reforma del Estatuto de Autonomía de Castilla La Mancha	
41012	El Euribor cae a su nivel más bajo en octubre	9
41013	Armstrong correrá el Giro 2009	2
41014	¿La penúltima vivienda de Obama?	1
41019	Rrecuperan cientos de piezas de gran valor	
41021	El novio de Nicole Richie se enfrenta a un pasajero en pleno vuelo	
41022	Jorge Garbajosa: "No me planteo renunciar a jugar con la selección española"	
41070	Detenido en Biarritz Zigor Goieskoetxea, miembro de Batasuna en Francia	3
41071	Las asociaciones piden que Mesquida deje la casa de la Guardia Civil donde sigue viviendo	
41072	Kundera delató al espía Miroslav Dvoracek ante la policía comunista de Praga en 1950	
41073	Condenado a 15 años el acusado de empujar a un hombre al Metro en Barcelona	
41074	Juan José Millás, Premio Nacional de Narrativa con la novela autobiográfica 'El Mundo'	
41085	Alonso reprueba las palabras de Rajoy ante el desfile del Día de la Hispanidad	4
41086	Chacón disculpa a Rajoy en su primer desfile como ministra	4
41087	Un edil vizcaíno del PSE, acusado de agresión sexual a su escolta	
41088	Puigcercós se impone en 9 de las12 federaciones de Esquerra	
41089	La manifestación ultraderechista de Tarragona acaba en fracaso	5
41090	Soldados españoles y afganos abaten a dos insurgentes al defender un convoy	
41091	El Rey aboga por una Europa bien coordinada ante la crisis	8
41092	El Govern inicia el proceso para dignificar a Companys	10
41093	El respeto a las instituciones	8
41094	REUNIÓN DEL G7 Frente occidental unido	
41095	CAMBIO CLIMÁTICO El clima ha cambiado	
41096	El milagro español del Santander y el BBVA	8
41097	CRISIS FINANCIERA Puesto en cuestión	8
41121	Un desfile no es fiesta	4
41122	¡Es la guerra, más madera!	
41123	A Dios rogando y con el mazo...	8
41124	Animus iocandi	
41125	Pasiones, premios y medallas	11
41126	Poder en rebeldía	
41127	Orinar en la calle	
41128	Cartas de los lectores I	
41129	Cartas de los lectores II	
41130	Violeta no tiene quien la defienda	
41131	Un arma de disuasión masiva	

41132	EDITORIAL: 'Normalidad en el 12-O'	5
41133	EDITORIAL: 'Operación confianza en la eurozona'	9
41134	La enfermedad es internet	
41173	El estadounidense Krugman, crítico del neoliberalismo, gana el Nobel de Economía	11
41174	Cervera (UPN) dice que no ha decantado su voto, pero adelanta que "estará a la altura"	7
41175	Santander saca pecho: ultima la compra de Sovereign e inyecta 1.200 millones en Abbey	
41176	Las familias pierden 130.000 millones: su patrimonio financiero vuelve al nivel del 92	8
41177	Un nuevo sondeo da a Obama una ventaja de diez puntos sobre John McCain	1
41178	El PSOE quiere sustituir a las diputadas de baja por maternidad	
41179	Fomento reconoce que la privatización del 30% de Aena va para largo	
41180	Un informe de FAES acusa al Gobierno de Zapatero de ser más belicista que el de Aznar	
41181	Se reducen a la mitad las contrataciones de fútbol en TV en pago por visión	
41182	iPod: el fin de una era	
41183	El fantasma de la crisis planeó sobre los corrillos del Palacio Real	8
41184	Los modestos también se aprietan el cinturón: los ejecutivos de FCC no volarán en business	
41185	El regalo de Telma a Letizia	
41186	Cuando las stars eran pobres	
41187	Miró ignora a su suegra	
41188	La indefensión de Jaime	
41189	Detenido en Biarritz Zigor Goieskoetxea, huido de la operación contra la cúpula de Batasuna	3
41190	Justicia tramitará la reparación y reconocimiento de la figura de Companys	10
41191	Las Fuerzas Armadas desfilan bajo el cielo gris de Madrid	4
41192	Los mossos ahogan una manifestación antifascista en Sants pero no evitan incidentes	5
41193	El PP acusa a Zapatero de ser "el presidente de los banqueros"	
41194	Catalunya ante el pasmo global	
41195	"Es hora de empujar"	
41196	Inicio inminente del proceso para lograr la nulidad del juicio a Companys	10
41197	Ultras y antifascistas se manifiestan sin problemas en Tarragona	5
41198	Chacón quita hierro a las palabras de Rajoy, porque "ese no es su sentimiento auténtico"	4
41199	El desfile aéreo se reduce debido a las condiciones meteorológicas en Madrid	4
41200	Desalojan a 60 jóvenes antifascistas encadenados en Montjuïc para boicotear hoy una concentración	5
41201	Chacón felicita vía videoconferencia el Día de la Hispanidad a distintas misiones españolas	4
41202	Por si vuelve Carpanta	

41208	Una parlamentaria del PP imputada en un caso de prevaricación	
41209	El PA expulsa a Carmona por flirtear con el PP	
41210	El PSOE pide al Ayuntamiento que actue en el proceso de Roca	
41211	Seis nominaciones para los premios turísticos World Travel Awards	
41212	Marbella busca fondos en Bruselas	
41213	La Junta decidirá la autorización al programa de Ayudas a Empresas Viables	
41214	Diputados propios para los inmigrantes	
41215	Teatro para El Vacie	
41216	Obama, 7 puntos por encima de McCain	1
41217	Lo último de Pedro J.: acusa ahora al catalán de provocar que un padre pueda perder la custodia de s	
41218	El Gobierno apoyará a la banca con un aval de 100.000 millones	8
41219	Losantos destapa a Rajoy: "Se han transmutado las rojigualdas por los coñazos"	4
41220	Rajoy es un señor normal	4
41221	Rajoy, del "feliz día de la nación" al "coñazo"	4
41222	Codicia	
41223	Había una vez un lobito bueno	
41224	Cristianismo maniqueo	
41225	Zapatero confirma que pondrá el plan de la UE en funcionamiento "de inmediato"	8
41281	La crisis económica y El Roto	8
41349	Los promotores inmobiliarios españoles ya no saben qué hacer para vender	8
41367	+ UPN será CIU	7
41384	Acto de inmigracion JSCoslada	
41412	“Todas las generaciones nacionalistas han actualizado su discurso, salvo ésta”	
41421	Comidas callejeras, hoteles, edificios y otras cosas de esos mundos viajeros de Dios	
41422	Murcia, en alerta por lluvias	6
41423	Las ayudas de la UE triplican las del plan de EEUU	8
41424	Aparecen manchas de hidrocarburo dispersas por la costa de Tarifa	
41425	Detienen a Zigor Goieskoetxea, huido de la operación contra la cúpula de Batasuna	3
41426	La pareja Rafa Nadal - Carlos Moyá debuta en el Masters Series de Madrid	
41427	El temporal apenas deja agua en las zonas de la Comunitat más afectadas por la sequía	6
41428	Lance Armstrong correrá el Giro 2009	2
41429	Detenidos en Valencia dos guardias civiles y un policía local por tráfico de drogas	
41430	Mugabe pasa por alto las negociaciones en Zimbabue y nombra a dos vicepresidentes	
41431	Cervera (UPN) dice que no ha decidido su voto, pero adelanta que "estará a la altura"	7
41432	El PP amenaza a UPN con una "ruptura	7

	unilateral" del pacto si no apoya su enmienda a los Presupuesto	
41433	Ibarretxe reitera que el País Vasco puede salir "reforzado" de la crisis	8
41434	Las bolsas reaccionan positivamente al plan de rescate de la Eurozona	9
41435	El polémico alcalde de Salamanca dejará la presidencia del PP regional	
41436	El vídeo de Madrid 2016	
41437	JST en el Comité Regional del PSC	
41438	Sacyr ajustará la plantilla de Vallehermoso y de sus oficinas centrales ante la crisis del sector	
41439	Espinosa informa a las CC.AA. de los avances de la dieta mediterránea como Patrimonio de la Humanida	
41440	Murcia vuelve a estar en alerta naranja ante la previsión de lluvias	6

A.2. Corpus del Experimento 2

El corpus con el que se trabaja en el experimento 2.2 se detalla a continuación.

Si el campo del cluster está vacío significa que el post representa en sí mismo un grupo temático con una sola noticia.

ID POST	TITULAR	CLUSTER
20783	Cuando el tenis se supera, nacen leyendas	1
20802	Futuro	
20816	Informarse a través de los medios: tarea imposible	2
20818	LA DIFICIL PAPELETA DEL SENADO	
20819	SACAPUNTAS (19)	
20844	La cobardía de José María Aznar	
20849	Luces y Sombras	2
20850	Una candidatura de unidad	

20851	Crisis de identidad	5
20852	No fue el día de Alonso	12
20854	España huele a cadáver	5
20855	Rosa Díez en la diana	
20861	Telecinco se convierte en la cadena con mayores ingresos por publicidad	17
20862	El Gorila Chavez suspende temporalmente su su programa "Aló Presidente"	
20863	Menos presupuesto y nuevos modelos para el periodismo de investigación en EEUU	
20864	El periodista que ha ganado la batalla al diario Sur de Málaga	
20874	Ortuzar asegura que el PNV acatará las reglas de juego si el TC suspende la consulta	
20883	Las primeras palabras de Zapatero a Pajín	15
20884	'No necesito el miembro para servir a España'	6
20885	Dos semanas de campaña por la lengua	14
20886	Nadal, a un palmo del número 1	1
20887	Al menos 40 muertos en un atentado en Kabul	11
20888	Peligro en el primer encierro de San Fermín	
20889	Sargento K	4
20890	Rock de alta intimidad	4
20891	Un nuevo sospechoso en el asesinato de dos jóvenes	
20892	Quince personas mueren en la carretera	3
20893	Unos 'héroes' a la sombra de Betancourt	2
20894	La deuda de las familias, principal obstáculo para sortear la crisis	5
20895	Windows, ¿qué estás haciendo?	
20896	Telecinco se hunde en el 'prime time'	17
20897	Los asientos del pasillo son los más seguros en un avión, según un estudio británico	
20898	Más colgado que un murciélago	
20899	Algo más que corbatas	
20900	José Antonio Alonso: "Nos vamos a tener que apretar el cinturón"	5
20901	El PP exige el cierre de la Central Nuclear de Cofrentes de manera "improrrogable"	
20902	De la portada de Playboy a los Juegos Olímpicos de Pekín	10
20903	China repartirá Biblias gratis con el logotipo de los Juegos Olímpicos en la solapa	10
20904	Al menos 28 muertos y decenas de heridos en Kabul por un atentado suicida	11
20906	Las portadas de la prensa se rinden a la hazaña de Nadal en Wimbledon	1
20907	Los Príncipes de Asturias volvieron a disfrutar de otro éxito español con Nadal	1
20908	Socialistas y populares estrenan sus nuevas ejecutivas con sendas reuniones en sus sedes	
20909	Street View se estrena a pedales en Europa	
20910	El Atlético va a por David Silva	
20911	Niegan la entrada al Ejército de un transexual por su "falta total de pene"	6
20912	Perú y Guatemala, a por México para lograr el	

	primer puesto en la lista de banderas	
20913	Saltan las alarmas: Cristiano Ronaldo puede estar KO tres meses	
20914	En el noreste llueve hasta en julio	9
20915	Detienen a un oscense por realizar tocamientos a una menor de 12 años	
20916	Alonso protesta contra la prensa: "Este ansia por el podio no es normal"	12
20917	La prensa inglesa tilda de "impresionante" la victoria del "Rey" Hamilton en Silverstone	12
20919	'Kyle XY' se enfrenta a 'CSI: Las Vegas' mientras busca su 'Identity'	
20920	Evacuadas cien personas por un espectacular incendio en el centro de Lisboa	
20925	Dieciséis muertos desde el comienzo de la primera operación salida del verano	3
20927	El Hornillo talará 4.000 pinos para pagar una deuda del Ayuntamiento	
20928	Laporta: "La moción de censura no ha triunfado y podemos agotar el mandato"	7
20931	Moda muy 'cool' para urbanitas	
20937	Medio centenar de alumnos no han cursado Educación para la Ciudadanía	
20940	Zapatero reúne a la nueva Comisión Ejecutiva del PSOE	15
20941	La semana comienza con lluvias en el norte del país	9
20942	Mueren 16 personas en accidentes de tráfico este fin de semana, cinco menos que en 2007	3
20943	Españoles hermanados con los «dálits»	
20944	Pedro Delgado: «Me siento agredido por la falta de respeto al español»	14
20945	La defensa de la escolarización en castellano irrumpe en el Parlamento europeo	
20946	Llegan a Tenerife 65 inmigrantes de un cayuco interceptado	
20947	Un hombre pierde la vida al estrellarse su avioneta en Murcia	
20948	La bodega del narco	
20949	Capos rusos comienzan a abandonar España para eludir sus detenciones	
20950	Rajoy asegura que la laicidad y el aborto «no le quitan el sueño a nadie»	16
20951	Culto al líder contra debate interno	15
20952	Los imputados de la trama de Estepona blanqueaban dinero en Marruecos	
20953	Jesús Caldera: «¿Una idea para la crisis? Para empezar, tranquilidad»	5
20954	Miguel Sebastián: «El debate no está en la corbata, sino en ahorrar energía» Miguel Sebastián _	15
20955	Camacho advierte a Nebrera de que no aceptará imposiciones	16

20956	Rajoy defiende a Camacho y llama al PPC a ser alternativa de Gobierno	16
20957	Dirigentes del PP acusan a Fernández de dar alas a la diputadacrítica	16
20958	Nebrera afirma que Rajoy será su único interlocutor	16
20959	Zapatero pide a España que crea en sí misma para superar la crisis	5
20960	La apuesta por los jóvenes marca la nueva ejecutiva	15
20961	El PSC espera que el 'giro plural' del PSOE facilite la financiación	15
20962	Sueños que se cumplen	
20978	El aborto	13
20979	Ya es el mejor de todos	1
20980	Hacer la cama	
20981	Gran Premio, gran retorno	12
20982	No fui a por la etapa	
20983	A pistola o a saeta	1
20984	«Goyistas» en tiempos de guerra	8
20985	Una explicación probable de la venta del oro	
20986	El burro de Goya	8
20987	De izquierdas y sin disfraces	15
20988	Nadal: El triunfo de la voluntad	1
20989	Histórico Rafa Nadal	1
20990	Zapatero se echa al monte	15
20991	Palpando progresismo	13
20992	El salario del miedo	
20993	El PSOE de la segunda legislatura	15
20994	Revolcón a Laporta	7
20995	Adiós y gracias	7
20996	Cisma en Canterbury	
20997	La influencia de Miguel de Unamuno	
20998	Madrid-Barcelona, como EEUU-UE	
20999	Hacia un Mohamed en la Generalitat	
21000	REENCUENTRO	

A.3. Corpus del Experimento 3

A continuación se muestra el corpus de noticias con el que se trabaja en el experimento 2.3.

Si el campo del cluster está vacío significa que el post representa en sí mismo un grupo temático con una sola noticia.

ID POST	TITULAR	CLUSTER
10040	Ya en Roma hubo noticia de Mariano	
10041	¿Qué les queda a las FARC?	
10042	¿Puede Obama ser presidente?	13
10162	Tracy Chapman	
10171	Contra la modificació de la Directiva Europea sobre Temps de Treball	3
10213	Cuando nadie actua, actua la ciudadanía	
10262	El PP defiende la jornada laboral de las 65 horas semanales	1
10355	Marcelino Iglesias: “Hemos conseguido que llueva”	4
10384	Premio al Esfuerzo Personal	
10408	Cuba: Transición	
10409	Perú: Guerra Sucia	
10442	Confesso que he estat piquet	
10539	+ Lección de "progresí" by Güemes	
10540	+ Los transportistas no son el problema, sino los impuestos	2
10541	+ Esperanza rules	6
10542	+ "Me gusta el país en el que vivimos... Me gusta la España desacelerada"	
10543	+ No hay calentamiento global este año	5
10544	+ Habla el director de Air Berlin	
10545	+ Tu chalet en primera línea de playa no corre peligro	5
10546	+ Los Bibis	
10562	El alcalde de Cuenca intenta justificar el enchufe de su hermano en el ayuntamiento	
10612	¿PP-UPV?	7
10613	Desaparecen compromisarios	
10677	Y Solbes cogió su fusil	
10678	Los viernes: Risoterapia (III)	
10679	Las notas de la Aguirre	6
10680	Esclavitud infantil	10
10724	Espe pierde los nervios	6
10768	75 anys fent adults *	
10769	Dissabte, Valors	
10805	Con violencia no hay acuerdo.	2
10830	La renta básica: Una base para el futuro	
10858	Política de desinformación o gobierno 1.0	
10859	¿65 horas? Ni de coña!!!!!	1
10905	Biblioteca: Plenilunio	
10906	Polvo en la obra	
10966	Ter, solidaritat o abús?	
11000	Arranca el Congreso Extraordinario del PSM	
11001	Sanidad Pública: ¡Presidenta!	6
11058	EL HOSPITAL DE LAS MENTIRAS	11
11059	EL MOSSAD TORRENTINO	11
11115	Chile y el machismo.	12
11116	Obama y Hillary	13

11117	TV machista y racista	12
11118	Peñafiel el machista .	12
11119	Capitostes y machismo	12
11120	Machismo en Argentina	12
11121	Metro de Madrid y campaña publicitaria.	
11122	Amaral en Anuncio de Expo Zaragoza 2008	4
11123	Greenpeace Parodia Anuncio de Dove	
11124	Sencillez y belleza.	
11125	Publicidad con violencia hacia el marido.	12
11126	Repsol versus Pestol	
11127	Simplón	
11128	El poder de Viagra.	
11130	Del esperpento a la locura..represiva	
11182	Algo de historia de Sevilla (by Oliseo28)	
11183	Más Paz	
11184	Un paseo por Sevilla (by ardid84)	8
11185	Sevilla : Paseo de la O (by defarol)	8
11186	Habitantes de una naranja	
11323	"UNA VEZ MÁS EL MAESTRO SABINA ME DEJA SIN PALABRAS"	
11324	"LA AUDIENCIA PROVINCIAL DE MURCIA CONDENA A LA EMPRESA DEL PROMOTOR JOSÉ LÓPEZ REJAS"	
11325	"ESPERANZA AGUIRRE PIERDE LOS NERVIOS Y MUESTRA SU VERDADERO TALANTE"	6
11409	Viernes	
11410	¿Cómo va todo?	
11493	¡Ni de coña!	1
11494	Cuando el poder se descalifica a si mismo	7
11607	Recuperando la normalidad	
11635	INCIATIVA PROPOSARÀ EN EL PARLAMENT VALENCIÀ UN POSICIONAMENT UNÀMIM FRONT LA JORNADA DE 65 HORES	3
11661	La noticia curiosa	
11690	No hacemos periodismo, fabricamos líderes.	
11691	65 horas…ni de coña!.	1
11692	Esperanza Aguirre entre abucheos	6
11693	Un último homenaje al gran maestro y ciudadano Gonzalo Anaya. Gracias por tu vida plena. In memoriam	
11795	Seguimos de fiesta. Gracias a todos, gracias Miguel	2
11820	OBAMA	13
11821	Ha muerto la Caja B ¡Vivan las Islas Caimán!	
11822	Desordenes y PSOE	2
11830	La esperanza irlandesa	
11851	CONGRESO COMARCAL	
11886	EKAIZER Y LAS PREJUBILACIONES EN EL PAÍS	
11887	PARA LA LIBERTAD: 40 AÑOS Y OTROS 25 MÁS	
11888	EDUCACIÓN, EXPLOTACIÓN, MÉXICO	10
11912	La Lider Esa	6
11913	Eduard Punset	
11914	¿Europa quiere ser neocañí?…	
11915	Vuelta a la infancia	
11931	La policía de Budapest prohíbe la marcha del Orgullo	9
11932	Fracasa la proyección del documental gay en el Parlamento italiano	9
11933	Celebración en Linares de aprobación en ayuntamiento de la moción contra la homofobia	9
11934	Santiago celebrará el Día del Orgullo Gay a partir del próximo lunes con cine	9
11935	Ayer se inauguró el Festival del Mar	9
11936	Amor sin nombre	9

11937	Polémica en Italia por un documental homosexual	9
11952	Más sobre campañas contra la directiva de las 65 horas	1
11974	Cafres	2

B. Tablas de resultados de los experimentos

Tabla de datos del EXPERIMENTO 2.1 (Para F-Measure con $\alpha = 0{,}2$ y $\alpha = 0{,}5$**)**

FMeasure_0.2_P-IP

	Ti 0	Ti 0,01	Ti 0,02	Ti 0,03	Ti 0,04	Ti 0,05	Ti 0,06	Ti 0,07	Ti 0,08	Ti 0,09	Ti 0,1	Ti 0,11	Ti 0,12	Ti 0,13	Ti 0,14	Ti 0,15	Ti 0,16	Ti 0,17	Ti 0,18	Ti 0,19	Ti 0,2
Tx 0	0,59	0,59	0,66	0,78	0,79	0,79	0,79	0,8	0,77	0,76	0,76	0,76	0,76	0,77	0,77	0,77	0,77	0,77	0,77	0,77	0,77
Tx 0,01	0,69	0,7	0,74	0,84	0,84	0,84	0,84	0,85	0,82	0,81	0,8	0,8	0,79	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8
Tx 0,02	0,74	0,75	0,81	0,87	0,86	0,86	0,85	0,85	0,83	0,82	0,82	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81
Tx 0,03	0,8	0,82	0,85	0,88	0,87	0,87	0,87	0,86	0,84	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,04	0,86	0,87	0,88	**0,91**	0,89	0,88	0,88	0,87	0,85	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,05	0,84	0,85	0,86	0,88	0,86	0,86	0,86	0,86	0,84	0,81	0,81	0,81	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8
Tx 0,06	0,83	0,85	0,86	0,87	0,86	0,86	0,85	0,86	0,83	0,81	0,8	0,8	0,8	0,79	0,78	0,78	0,78	0,78	0,78	0,78	0,78
Tx 0,07	0,83	0,85	0,86	0,87	0,86	0,86	0,85	0,86	0,83	0,81	0,8	0,8	0,8	0,79	0,78	0,78	0,78	0,78	0,78	0,78	0,78
Tx 0,08	0,82	0,83	0,84	0,84	0,83	0,83	0,82	0,81	0,8	0,79	0,78	0,78	0,77	0,77	0,76	0,76	0,76	0,76	0,76	0,76	0,76
Tx 0,09	0,82	0,83	0,84	0,84	0,83	0,83	0,82	0,81	0,8	0,78	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,1	0,82	0,83	0,84	0,84	0,83	0,83	0,82	0,81	0,8	0,78	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,11	0,82	0,83	0,84	0,84	0,83	0,83	0,82	0,81	0,8	0,78	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,12	0,82	0,83	0,83	0,83	0,82	0,82	0,81	0,8	0,79	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,13	0,82	0,83	0,83	0,83	0,82	0,82	0,81	0,8	0,79	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,14	0,82	0,83	0,83	0,83	0,82	0,82	0,81	0,8	0,79	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,15	0,82	0,83	0,83	0,83	0,82	0,82	0,81	0,8	0,79	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,16	0,82	0,83	0,83	0,83	0,82	0,82	0,81	0,8	0,79	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,17	0,82	0,83	0,83	0,83	0,82	0,82	0,81	0,8	0,79	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,18	0,82	0,83	0,83	0,83	0,82	0,82	0,81	0,8	0,79	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,19	0,82	0,83	0,83	0,83	0,82	0,82	0,81	0,8	0,79	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,2	0,82	0,83	0,83	0,83	0,82	0,82	0,81	0,8	0,79	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75

FMeasure_0.5_P-IP

	Ti 0	Ti 0,01	Ti 0,02	Ti 0,03	Ti 0,04	Ti 0,05	Ti 0,06	Ti 0,07	Ti 0,08	Ti 0,09	Ti 0,1	Ti 0,11	Ti 0,12	Ti 0,13	Ti 0,14	Ti 0,15	Ti 0,16	Ti 0,17	Ti 0,18	Ti 0,19	Ti 0,2
Tx 0	0,36	0,36	0,44	0,61	0,64	0,64	0,64	0,65	0,64	0,64	0,64	0,64	0,64	0,65	0,66	0,66	0,66	0,66	0,66	0,66	0,66
Tx 0,01	0,47	0,48	0,54	0,69	0,72	0,73	0,73	0,73	0,72	0,72	0,72	0,72	0,72	0,73	0,74	0,74	0,74	0,74	0,74	0,74	0,74
Tx 0,02	0,54	0,56	0,65	0,78	0,78	0,79	0,79	0,8	0,79	0,79	0,78	0,78	0,78	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79
Tx 0,03	0,65	0,69	0,77	0,84	0,84	0,84	0,85	0,85	0,84	0,84	0,83	0,83	0,83	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84
Tx 0,04	0,76	0,79	0,84	**0,89**	**0,89**	**0,89**	**0,89**	**0,89**	0,88	0,86	0,86	0,86	0,86	0,87	0,86	0,86	0,86	0,86	0,86	0,86	0,86
Tx 0,05	0,76	0,78	0,83	0,88	0,87	0,88	0,88	0,88	0,87	0,86	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85
Tx 0,06	0,76	0,79	0,84	0,88	0,87	0,88	0,88	**0,89**	0,87	0,86	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85
Tx 0,07	0,76	0,79	0,84	0,88	0,87	0,88	0,88	**0,89**	0,87	0,86	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85
Tx 0,08	0,75	0,78	0,83	0,86	0,85	0,86	0,86	0,86	0,85	0,84	0,84	0,84	0,84	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83
Tx 0,09	0,75	0,78	0,83	0,86	0,85	0,86	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83
Tx 0,1	0,75	0,78	0,83	0,86	0,85	0,86	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83
Tx 0,11	0,75	0,78	0,83	0,86	0,85	0,86	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83
Tx 0,12	0,75	0,78	0,82	0,85	0,85	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,13	0,75	0,78	0,82	0,85	0,85	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,14	0,75	0,78	0,82	0,85	0,85	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,15	0,75	0,78	0,82	0,85	0,85	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,16	0,75	0,78	0,82	0,85	0,85	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,17	0,75	0,78	0,82	0,85	0,85	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,18	0,75	0,78	0,82	0,85	0,85	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,19	0,75	0,78	0,82	0,85	0,85	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,2	0,75	0,78	0,82	0,85	0,85	0,85	0,85	0,85	0,84	0,83	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82

*** En rojo se muestran los valores más altos de F-Measure para los umbrales determinados.**

Tabla de datos del EXPERIMENTO 2.2 (Para F-Measure con $\alpha = 0{,}2$ y $\alpha = 0{,}5$**)**

FMeasure_0.2_P-IP

	Ti 0	Ti 0,01	Ti 0,02	Ti 0,03	Ti 0,04	Ti 0,05	Ti 0,06	Ti 0,07	Ti 0,08	Ti 0,09	Ti 0,1	Ti 0,11	Ti 0,12	Ti 0,13	Ti 0,14	Ti 0,15	Ti 0,16	Ti 0,17	Ti 0,18	Ti 0,19	Ti 0,2
Tx 0	0,58	0,58	0,67	0,7	0,71	0,74	0,74	0,74	0,74	0,74	0,75	0,74	0,74	0,74	0,73	0,73	0,73	0,73	0,73	0,73	0,73
Tx 0,01	0,7	0,7	0,74	0,78	**0,79**	**0,79**	**0,79**	**0,79**	**0,79**	0,77	0,78	0,77	0,77	0,77	0,76	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,02	0,75	0,75	0,78	**0,79**	0,77	0,78	0,78	0,78	0,77	0,78	0,78	0,77	0,77	0,77	0,76	0,76	0,76	0,76	0,76	0,76	0,76
Tx 0,03	0,77	0,77	**0,79**	**0,79**	0,78	0,78	0,78	0,78	0,77	0,77	0,78	0,77	0,76	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,04	0,78	0,78	**0,79**	0,78	0,77	0,77	0,77	0,77	0,76	0,76	0,77	0,76	0,75	0,75	0,74	0,74	0,74	0,74	0,74	0,73	0,73
Tx 0,05	0,77	0,77	0,78	0,77	0,76	0,76	0,76	0,76	0,75	0,75	0,75	0,74	0,73	0,73	0,72	0,73	0,73	0,73	0,73	0,72	0,72
Tx 0,06	0,77	0,77	0,78	0,77	0,76	0,76	0,76	0,76	0,74	0,74	0,74	0,73	0,73	0,73	0,72	0,72	0,72	0,72	0,72	0,71	0,71
Tx 0,07	0,77	0,77	0,78	0,77	0,76	0,75	0,75	0,75	0,74	0,73	0,73	0,72	0,72	0,72	0,71	0,71	0,71	0,71	0,71	0,7	0,7
Tx 0,08	0,77	0,77	0,78	0,77	0,75	0,74	0,74	0,74	0,73	0,72	0,72	0,72	0,71	0,71	0,7	0,7	0,7	0,7	0,7	0,69	0,69
Tx 0,09	0,76	0,76	0,76	0,76	0,73	0,73	0,73	0,73	0,71	0,71	0,71	0,7	0,69	0,69	0,68	0,68	0,68	0,68	0,68	0,67	0,67
Tx 0,1	0,76	0,76	0,76	0,76	0,73	0,73	0,73	0,73	0,71	0,71	0,71	0,7	0,69	0,69	0,68	0,68	0,68	0,68	0,68	0,67	0,67
Tx 0,11	0,76	0,76	0,76	0,76	0,73	0,73	0,73	0,73	0,71	0,71	0,71	0,7	0,69	0,69	0,68	0,68	0,68	0,68	0,68	0,67	0,67
Tx 0,12	0,76	0,76	0,76	0,76	0,73	0,73	0,73	0,73	0,71	0,71	0,71	0,7	0,69	0,69	0,68	0,68	0,68	0,68	0,68	0,67	0,67
Tx 0,13	0,76	0,76	0,76	0,76	0,73	0,73	0,73	0,73	0,71	0,71	0,71	0,7	0,69	0,69	0,68	0,68	0,68	0,68	0,68	0,67	0,67
Tx 0,14	0,76	0,76	0,76	0,76	0,73	0,73	0,73	0,73	0,71	0,71	0,71	0,7	0,69	0,69	0,68	0,68	0,68	0,68	0,68	0,67	0,67
Tx 0,15	0,75	0,75	0,76	0,75	0,73	0,72	0,72	0,72	0,7	0,7	0,7	0,69	0,68	0,68	0,67	0,67	0,67	0,67	0,67	0,66	0,66
Tx 0,16	0,75	0,75	0,76	0,75	0,73	0,72	0,72	0,72	0,7	0,7	0,7	0,69	0,68	0,68	0,67	0,67	0,67	0,67	0,67	0,66	0,66
Tx 0,17	0,75	0,75	0,76	0,75	0,73	0,72	0,72	0,72	0,7	0,7	0,7	0,69	0,68	0,68	0,67	0,67	0,67	0,67	0,67	0,66	0,66
Tx 0,18	0,75	0,75	0,76	0,75	0,73	0,72	0,72	0,72	0,7	0,7	0,7	0,69	0,68	0,68	0,67	0,67	0,67	0,67	0,67	0,66	0,66
Tx 0,19	0,75	0,75	0,76	0,75	0,73	0,72	0,72	0,72	0,7	0,7	0,7	0,69	0,68	0,68	0,67	0,67	0,67	0,67	0,67	0,66	0,66
Tx 0,2	0,75	0,75	0,76	0,75	0,73	0,72	0,72	0,72	0,7	0,7	0,7	0,69	0,68	0,68	0,67	0,67	0,67	0,67	0,67	0,66	0,66

FMeasure_0.5_P-IP

	Ti 0	Ti 0,01	Ti 0,02	Ti 0,03	Ti 0,04	Ti 0,05	Ti 0,06	Ti 0,07	Ti 0,08	Ti 0,09	Ti 0,1	Ti 0,11	Ti 0,12	Ti 0,13	Ti 0,14	Ti 0,15	Ti 0,16	Ti 0,17	Ti 0,18	Ti 0,19	Ti 0,2
Tx 0	0,36	0,36	0,46	0,5	0,51	0,58	0,58	0,59	0,6	0,62	0,64	0,63	0,64	0,64	0,64	0,64	0,64	0,64	0,64	0,64	0,64
Tx 0,01	0,49	0,49	0,57	0,63	0,63	0,68	0,68	0,68	0,69	0,7	0,71	0,71	0,71	0,71	0,71	0,71	0,71	0,71	0,71	0,71	0,71
Tx 0,02	0,63	0,63	0,69	0,73	0,74	0,76	0,76	0,77	0,77	0,79	0,8	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79
Tx 0,03	0,71	0,71	0,75	0,77	0,77	0,79	0,79	0,79	0,8	**0,82**	**0,82**	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81
Tx 0,04	0,73	0,73	0,77	0,79	0,78	0,8	0,8	0,8	0,8	**0,82**	**0,82**	**0,82**	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81
Tx 0,05	0,74	0,74	0,77	0,79	0,79	0,8	0,8	0,8	0,8	0,81	0,81	0,81	0,81	0,81	0,8	0,8	0,8	0,8	0,8	0,8	0,8
Tx 0,06	0,75	0,75	0,78	0,8	0,79	0,8	0,8	0,8	0,8	0,81	0,81	0,81	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,79	0,79
Tx 0,07	0,75	0,75	0,78	0,8	0,79	0,8	0,8	0,8	0,8	0,8	0,81	0,8	0,8	0,8	0,79	0,79	0,79	0,79	0,79	0,78	0,78
Tx 0,08	0,75	0,75	0,78	0,8	0,78	0,79	0,79	0,8	0,79	0,8	0,8	0,79	0,79	0,79	0,78	0,78	0,78	0,78	0,78	0,78	0,78
Tx 0,09	0,74	0,74	0,77	0,79	0,77	0,78	0,78	0,78	0,78	0,78	0,79	0,78	0,77	0,77	0,77	0,77	0,77	0,77	0,77	0,76	0,76
Tx 0,1	0,74	0,74	0,77	0,79	0,77	0,78	0,78	0,78	0,78	0,78	0,79	0,78	0,77	0,77	0,77	0,77	0,77	0,77	0,77	0,76	0,76
Tx 0,11	0,74	0,74	0,77	0,79	0,77	0,78	0,78	0,78	0,78	0,78	0,79	0,78	0,77	0,77	0,77	0,77	0,77	0,77	0,77	0,76	0,76
Tx 0,12	0,74	0,74	0,77	0,79	0,77	0,78	0,78	0,78	0,78	0,78	0,79	0,78	0,77	0,77	0,77	0,77	0,77	0,77	0,77	0,76	0,76
Tx 0,13	0,74	0,74	0,77	0,79	0,77	0,78	0,78	0,78	0,78	0,78	0,79	0,78	0,77	0,77	0,77	0,77	0,77	0,77	0,77	0,76	0,76
Tx 0,14	0,74	0,74	0,77	0,79	0,77	0,78	0,78	0,78	0,78	0,78	0,79	0,78	0,77	0,77	0,77	0,77	0,77	0,77	0,77	0,76	0,76
Tx 0,15	0,73	0,73	0,76	0,78	0,77	0,77	0,77	0,78	0,77	0,78	0,78	0,77	0,77	0,77	0,76	0,76	0,76	0,76	0,76	0,76	0,76
Tx 0,16	0,73	0,73	0,76	0,78	0,77	0,77	0,77	0,78	0,77	0,78	0,78	0,77	0,77	0,77	0,76	0,76	0,76	0,76	0,76	0,76	0,76
Tx 0,17	0,73	0,73	0,76	0,78	0,77	0,77	0,77	0,78	0,77	0,78	0,78	0,77	0,77	0,77	0,76	0,76	0,76	0,76	0,76	0,76	0,76
Tx 0,18	0,73	0,73	0,76	0,78	0,77	0,77	0,77	0,78	0,77	0,78	0,78	0,77	0,77	0,77	0,76	0,76	0,76	0,76	0,76	0,76	0,76
Tx 0,19	0,73	0,73	0,76	0,78	0,77	0,77	0,77	0,78	0,77	0,78	0,78	0,77	0,77	0,77	0,76	0,76	0,76	0,76	0,76	0,76	0,76
Tx 0,2	0,73	0,73	0,76	0,78	0,77	0,77	0,77	0,78	0,77	0,78	0,78	0,77	0,77	0,77	0,76	0,76	0,76	0,76	0,76	0,76	0,76

*** En rojo se muestran los valores más altos de F-Measure para los umbrales determinados.**

Tabla de datos del EXPERIMENTO 2.3 (Para F-Measure con $\alpha = 0{,}2$ y $\alpha = 0{,}5$**)**

F-Measure_0.2_P-IP

	Ti 0	Ti 0,01	Ti 0,02	Ti 0,03	Ti 0,04	Ti 0,05	Ti 0,06	Ti 0,07	Ti 0,08	Ti 0,09	Ti 0,10	Ti 0,11	Ti 0,12	Ti 0,13	Ti 0,14	Ti 0,15	Ti 0,16	Ti 0,17	Ti 0,18	Ti 0,19	Ti 0,20	Ti 0,21	Ti 0,22	Ti 0,23	Ti 0,24	Ti 0,25	Ti 0,26	Ti 0,27	Ti 0,28	Ti 0,29	Ti 0,3
Tx 0	0,53	0,53	0,53	0,55	0,55	0,56	0,59	0,59	0,61	0,61	0,61	0,61	0,61	0,65	0,68	0,68	0,68	0,68	0,68	0,73	0,74	0,74	0,73	0,73	0,73	0,73	0,73	0,73	0,73	0,73	0,73
Tx 0,01	0,62	0,62	0,62	0,65	0,65	0,67	0,69	0,69	0,7	0,7	0,7	0,7	0,7	0,73	0,76	0,76	0,76	0,76	0,76	0,8	0,81	0,8	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79
Tx 0,02	0,68	0,68	0,68	0,77	0,77	0,78	0,8	0,8	0,81	0,81	0,81	0,81	0,81	0,82	0,83	0,83	0,83	0,83	0,83	0,85	0,86	0,85	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83
Tx 0,03	0,83	0,83	0,83	0,85	0,86	0,87	0,87	0,87	0,88	0,88	0,88	0,88	0,88	0,89	0,89	0,89	0,89	0,89	0,89	0,89	0,89	0,89	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87
Tx 0,04	0,87	0,87	0,87	0,88	0,88	0,89	0,89	0,89	0,89	0,9	0,9	0,9	0,9	0,9	0,89	0,89	0,89	0,89	0,89	0,89	0,89	0,89	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86
Tx 0,05	0,87	0,87	0,87	0,89	0,89	0,89	0,89	0,89	0,89	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,9	0,89	0,89	0,89	0,89	0,89	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86
Tx 0,06	0,85	0,85	0,85	0,86	0,85	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,87	0,86	0,86	0,86	0,85	0,85	0,86	0,86	0,85	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,07	0,85	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,86	0,86	0,87	0,87	0,85	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82
Tx 0,08	0,85	0,85	0,85	0,86	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,87	0,86	0,86	0,86	0,85	0,85	0,86	0,86	0,84	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,82	0,81
Tx 0,09	0,84	0,84	0,84	0,85	0,85	0,85	0,85	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,85	0,85	0,85	0,84	0,84	0,85	0,85	0,83	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,8
Tx 0,10	0,84	0,84	0,84	0,85	0,85	0,85	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,85	0,85	0,84	0,84	0,83	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,8
Tx 0,11	0,84	0,84	0,84	0,85	0,85	0,85	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,85	0,85	0,84	0,84	0,83	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,8
Tx 0,12	0,84	0,84	0,84	0,85	0,85	0,85	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,85	0,85	0,84	0,84	0,83	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,8
Tx 0,13	0,84	0,84	0,84	0,85	0,85	0,85	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,85	0,85	0,84	0,84	0,83	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,8
Tx 0,14	0,84	0,84	0,84	0,85	0,85	0,85	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,85	0,85	0,84	0,84	0,83	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,81	0,8
Tx 0,15	0,83	0,83	0,83	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,83	0,83	0,83	0,83	0,81	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,78
Tx 0,16	0,83	0,83	0,83	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,83	0,83	0,83	0,83	0,81	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,78
Tx 0,17	0,83	0,83	0,83	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,83	0,83	0,83	0,83	0,81	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,78
Tx 0,18	0,83	0,83	0,83	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,84	0,83	0,83	0,83	0,83	0,81	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,79	0,78
Tx 0,19	0,82	0,82	0,82	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,84	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,8	0,78	0,78	0,78	0,78	0,78	0,78	0,78	0,78	0,77
Tx 0,20	0,82	0,82	0,82	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,83	0,84	0,83	0,83	0,83	0,82	0,82	0,82	0,82	0,8	0,78	0,78	0,78	0,78	0,78	0,78	0,78	0,78	0,77

F-Measure_0.5_P-IP

	Ti 0	Ti 0,01	Ti 0,02	Ti 0,03	Ti 0,04	Ti 0,05	Ti 0,06	Ti 0,07	Ti 0,08	Ti 0,09	Ti 0,10	Ti 0,11	Ti 0,12	Ti 0,13	Ti 0,14	Ti 0,15	Ti 0,16	Ti 0,17	Ti 0,18	Ti 0,19	Ti 0,20	Ti 0,21	Ti 0,22	Ti 0,23	Ti 0,24	Ti 0,25	Ti 0,26	Ti 0,27	Ti 0,28	Ti 0,29	Ti 0,3
Tx 0	0,31	0,31	0,31	0,33	0,33	0,34	0,37	0,37	0,38	0,38	0,38	0,38	0,38	0,42	0,47	0,47	0,47	0,47	0,47	0,54	0,56	0,55	0,55	0,55	0,55	0,55	0,55	0,55	0,55	0,55
Tx 0,01	0,4	0,4	0,4	0,43	0,43	0,45	0,47	0,47	0,48	0,48	0,48	0,48	0,48	0,52	0,59	0,59	0,59	0,59	0,59	0,64	0,66	0,66	0,65	0,65	0,65	0,65	0,65	0,65	0,65	0,65
Tx 0,02	0,47	0,47	0,47	0,59	0,59	0,6	0,64	0,64	0,65	0,65	0,65	0,65	0,65	0,68	0,71	0,71	0,71	0,71	0,71	0,75	0,76	0,76	0,75	0,75	0,75	0,75	0,75	0,75	0,75	0,75
Tx 0,03	0,7	0,7	0,7	0,74	0,77	0,77	0,79	0,79	0,8	0,8	0,8	0,8	0,8	0,82	0,84	0,84	0,84	0,84	0,84	0,87	0,87	0,87	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86
Tx 0,04	0,79	0,79	0,79	0,83	0,83	0,84	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,87	0,88	0,88	0,88	0,87	0,87	0,89	0,88	0,89	0,88	0,88	0,88	0,88	0,88	0,88	0,88	0,88
Tx 0,05	0,81	0,81	0,81	0,85	0,85	0,86	0,87	0,87	0,88	0,88	0,88	0,88	0,88	0,89	0,89	0,9	0,9	0,9	0,9	0,91	0,88	0,91	0,89	0,89	0,89	0,89	0,89	0,89	0,89	0,89
Tx 0,06	0,8	0,8	0,8	0,83	0,83	0,84	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,87	0,87	0,87	0,87	0,88	0,89	0,89	0,88	0,88	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86
Tx 0,07	0,82	0,82	0,82	0,84	0,85	0,85	0,86	0,86	0,87	0,87	0,87	0,87	0,87	0,88	0,88	0,88	0,88	0,88	0,88	0,89	0,88	0,89	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87
Tx 0,08	0,82	0,82	0,82	0,85	0,85	0,85	0,86	0,86	0,87	0,87	0,87	0,87	0,87	0,88	0,88	0,88	0,88	0,88	0,88	0,89	0,88	0,89	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87
Tx 0,09	0,82	0,82	0,82	0,85	0,85	0,85	0,86	0,86	0,87	0,87	0,87	0,87	0,87	0,88	0,88	0,88	0,88	0,87	0,87	0,89	0,88	0,88	0,86	0,86	0,86	0,86	0,86	0,86	0,86	0,86
Tx 0,10	0,82	0,82	0,82	0,85	0,85	0,86	0,87	0,87	0,87	0,88	0,88	0,88	0,88	0,89	0,88	0,88	0,88	0,88	0,88	0,89	0,88	0,88	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87
Tx 0,11	0,82	0,82	0,82	0,85	0,85	0,86	0,87	0,87	0,87	0,88	0,88	0,88	0,88	0,89	0,88	0,88	0,88	0,88	0,88	0,89	0,88	0,88	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87
Tx 0,12	0,82	0,82	0,82	0,85	0,85	0,86	0,87	0,87	0,87	0,88	0,88	0,88	0,88	0,89	0,88	0,88	0,88	0,88	0,88	0,89	0,87	0,88	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87
Tx 0,13	0,82	0,82	0,82	0,85	0,85	0,86	0,87	0,87	0,87	0,88	0,88	0,88	0,88	0,89	0,88	0,88	0,88	0,88	0,88	0,89	0,87	0,88	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87
Tx 0,14	0,82	0,82	0,82	0,85	0,85	0,86	0,87	0,87	0,87	0,88	0,88	0,88	0,88	0,89	0,88	0,88	0,88	0,88	0,88	0,89	0,87	0,88	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87
Tx 0,15	0,82	0,82	0,82	0,85	0,85	0,85	0,86	0,86	0,86	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,86	0,87	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85
Tx 0,16	0,82	0,82	0,82	0,85	0,85	0,85	0,86	0,86	0,86	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,86	0,87	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85
Tx 0,17	0,82	0,82	0,82	0,85	0,85	0,85	0,86	0,86	0,86	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,86	0,87	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85
Tx 0,18	0,82	0,82	0,82	0,85	0,85	0,85	0,86	0,86	0,86	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,87	0,86	0,87	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85

Tx 0,19	0,82	0,82	0,82	0,84	0,85	0,85	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,87	0,87	0,87	0,87	0,86	0,86	0,86	0,86	0,86	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,84
Tx 0,20	0,82	0,82	0,82	0,84	0,85	0,85	0,85	0,85	0,86	0,86	0,86	0,86	0,86	0,87	0,87	0,87	0,87	0,86	0,86	0,86	0,86	0,86	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,85	0,84

*** En rojo se muestran los valores más altos de F-Measure para los umbrales determinado.**